

Modeling with ODE

P. Howard

Spring 2005

Contents

1	Overview	2
2	Modeling with ODE	2
2.1	Compartment Analysis	2
2.2	Chemical Reactions	4
2.2.1	Elementary Reactions	4
2.2.2	Complex Reactions	4
2.2.3	Rates of reaction	5
2.2.4	Determining Reaction Rates	6
2.2.5	Carbon Dating	7
2.3	Population Dynamics	8
2.4	Newtonian mechanics.	12
2.4.1	Newtonian Mechanics in Polar Coordinates	14
2.5	Hamiltonian Mechanics	15
2.6	Variational Methods	16
3	Well-posedness Theory	17
3.1	Stability Theory	21
3.2	Stability and Eigenvalues	25
3.3	Maximum Sustainable Yield	29
3.4	Uniqueness Theory	30
3.5	Existence Theory	33
4	Solving Ordinary Differential Equations in MATLAB	37
4.1	Finding Explicit Solutions	37
4.1.1	First Order Equations	37
4.1.2	Second and Higher Order Equations	38
4.1.3	Systems	38
4.2	Finding Numerical Solutions	39
4.2.1	First Order Equations	39
4.2.2	Second Order Equations	40
4.2.3	Solving Systems of ODE	41
4.3	Laplace Transforms	42
4.4	Boundary Value Problems	42
4.5	Event Location	43
5	Numerical Methods	44
5.1	Euler's Method	45
6	Advanced ODE Solvers	45
6.1	Stiff ODE	46

1 Overview

A wide variety of natural phenomena such as projectile motion, the flow of electric current, and the progression of chemical reactions are well described by equations that relate changing quantities. As the derivative of a function provides the rate at which that function is changing with respect to its independent variable, the equations describing these phenomena often involve one or more derivatives, and we refer to them as differential equations. In these notes we consider three critical aspects in the theory of ordinary differential equations: 1. Developing models of physical phenomena, 2. Determining whether our models are mathematically “well-posed” (do solutions exist? are these solutions unique? do the solutions we find for our equation genuinely correspond with the phenomenon we are modeling), and 3. Solving ODE numerically with MATLAB.

2 Modeling with ODE

Solutions to ordinary differential equations cannot be determined uniquely without some outside condition, typically an initial value or a boundary value. In order to understand the nature of this information, consider the general first order equation

$$y' = f(t, y), \tag{2.1}$$

for which $'$ denotes differentiation with respect to t . Assuming $f(t, y)$ is sufficiently differentiable, we can develop a solution to (2.1) for t sufficiently small through the Taylor approximation,

$$y(t) = y(0) + y'(0)t + \frac{1}{2}y''(0)t^2 + \dots \tag{2.2}$$

Suppose we know the initial value for $y(t)$, $y(0)$. Observe, then, that we can compute $y'(0)$ directly from (2.1):

$$y'(0) = f(0, y(0)).$$

Similarly, by differentiating (2.1) with respect to t , we have

$$y'' = \frac{\partial}{\partial t}f(t, y) + \frac{\partial}{\partial y}f(t, y)y',$$

and we can compute $y''(0)$ as

$$y''(0) = \frac{\partial}{\partial t}f(0, y(0)) + \frac{\partial}{\partial y}f(0, y(0))y'(0).$$

Proceeding similarly, we can develop the entirety of expansion (2.2).

2.1 Compartment Analysis

Suppose $x(t)$ represents the amount of substance in some compartment at time t . For example, $x(t)$ might represent the liters of gasoline in a particular tank or the grams of medicine in a particular organ. We can represent the change in quantity $x(t)$ in terms of the amount of this quantity flowing into the compartment and the amount flowing out, as

$$\frac{dx}{dt} = \text{input rate} - \text{output rate}.$$

Example 2.1. (Drug concentration in an organ.) Suppose blood carries a certain drug into an organ at variable rate $r_I(t)$ cm³/s and out of the organ with variable rate $r_O(t)$ cm³/s, and that the organ has an initial blood volume V cm³. If the concentration of blood entering the organ is $c(t)$ g/cm³, determine an ODE for the amount of drug in the organ at time t .

Let $x(t)$ denote the amount of drug in the organ at time t , measured in grams. The input rate is then $r_I(t)c(t)$, while the output rate, assuming instantaneous mixing, is $\frac{x(t)}{V(t)}r_O(t)$, where the volume of blood in

the organ $V(t)$ can be computed as the initial volume V plus the difference between the blood that flows into the organ over time t and the blood that flows out during the same time:

$$V(t) = V + \int_0^t r_I(s) - r_O(s) ds.$$

We have, then, the ODE

$$\frac{dx}{dt} = r_I(t)c(t) - \frac{r_O(t)}{V + \int_0^t r_I(s) - r_O(s) ds} x(t) = 0.$$

△

Example 2.2. (Cleaning the Great Lakes.) The Great Lakes are connected by a network of waterways, as roughly depicted in Figure 2.1. Assume the volume of each lake remains constant, and that water flows into the lake with volume V_k at rate r_k . Suppose pollution stops abruptly (i.e., no more pollution flows into the lakes) and develop a system of ODE that models the progression of pollution as it clears from the lakes.

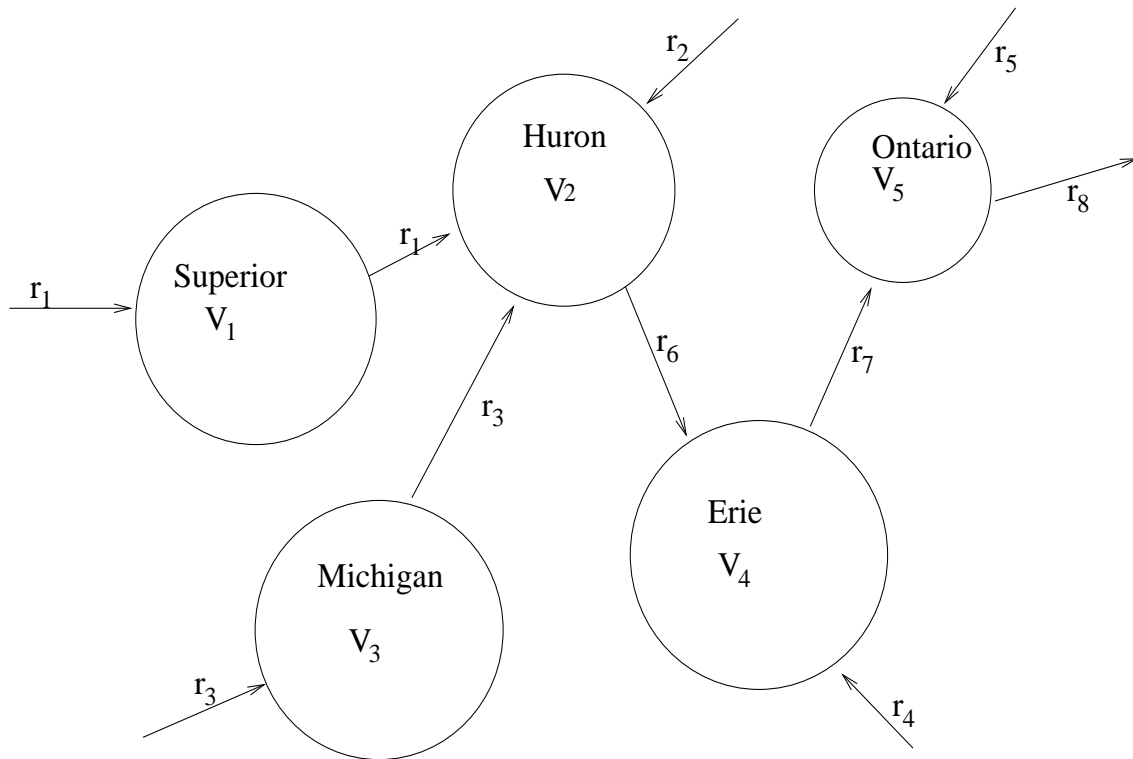


Figure 2.1: Great Lakes waterway network.

We have one differential equation for each lake. Let $x_k(t)$ represent the amount of pollutant in the lake

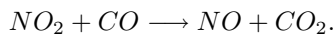
with volume V_k . We obtain the system

$$\begin{aligned}\frac{dx_1}{dt} &= -\frac{r_1}{V_1}x_1 \\ \frac{dx_2}{dt} &= +\frac{r_1}{V_1}x_1 + \frac{r_3}{V_3}x_3 - \frac{r_1 + r_2 + r_3}{V_2}x_2 \\ \frac{dx_3}{dt} &= -\frac{r_3}{V_3}x_3 \\ \frac{dx_4}{dt} &= \frac{r_1 + r_2 + r_3}{V_2}x_2 - \frac{r_1 + r_2 + r_3 + r_4}{V_4}x_4 \\ \frac{dx_5}{dt} &= \frac{r_1 + r_2 + r_3 + r_4}{V_4}x_4 - \frac{r_1 + r_2 + r_3 + r_4 + r_5}{V_5}x_5.\end{aligned}$$

△

2.2 Chemical Reactions

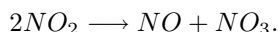
A chemical reaction is a process of molecular rearrangement by which one or more substances may be transformed into one or more new substances. For example, nitrogen dioxide (chemical symbol NO_2) reacts with carbon monoxide (CO) to form nitrogen monoxide and carbon dioxide,



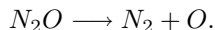
Energy is released or absorbed in a chemical reaction, but no change in total molecular weight occurs.

2.2.1 Elementary Reactions

Elementary reactions (sometimes referred to as *simple* reactions) are the building blocks of chemical reactions. They are reactions for which no intermediate steps occur. For example, the reaction of nitrogen oxide with itself to form nitrogen monoxide and nitrogen trioxide is elementary. In this case, the *stoichiometric* equation (i.e., the balanced reaction equation) has the form



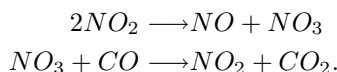
In particular, this is an example of a second order elementary reaction, because two molecules interact (NO_2 with itself). (A first order reaction, though not elementary, is



Higher order reactions are uncommon.¹⁾

2.2.2 Complex Reactions

In most cases, several steps occur in the mechanism by which initial reactants form their products. In the reaction above between nitrogen dioxide and carbon monoxide, one proposed mechanism consists of the following elementary steps:

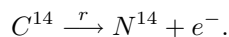


The designation “proposed” mechanism is a recognition that it is extremely difficult to know with certainty what is happening at the intermolecular level in a given reaction. In general, the best we can do is propose a mechanism that fits all known experimental data.

¹Which is to say, I couldn't hunt one down to use as an example.

2.2.3 Rates of reaction

Recall that for radioactive decay, the decrease in decaying substance is assumed proportional to the amount left. For example, the process of carbon dating depends on the decay of carbon-14 (a carbon isotope with six protons and eight neutrons) into nitrogen-14, in which an electron is released,



The assumption that the rate of decay is proportional to the amount left, can be written in the form

$$\frac{d[C^{14}]}{dt} = -r[C^{14}],$$

where $[C^{14}]$ represents the concentration of carbon-14 at time t , typically measured as moles per unit volume. (Recall that 1 mole is approximately 6.024×10^{23} molecules, where 6.024×10^{23} is roughly Avogadro's number, which corresponds with the number of atoms in a 12 gram sample of carbon-12.) According to the conservation of mass, we must have

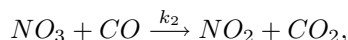
$$\frac{d[N^{14}]}{dt} = +r[C^{14}],$$

which is simply to say that for each molecule of carbon-14 lost, a molecule of nitrogen-14 is gained.

In general, for elementary reactions, we will assume the *law of mass action*.

Law of mass action. The rate of a chemical reaction is proportion to the product of the concentrations of the reactants.

In the case of our elementary reaction above between nitrogen trioxide and carbon monoxide,



the law of mass action asserts,

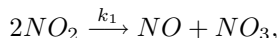
$$\frac{d[NO_3]}{dt} = \frac{d[CO]}{dt} = -k_2[NO_3][CO].$$

A good intuitive way to think about this is that since the nitrogen trioxide and the carbon monoxide only react when they come into contact with one another, the chance of reaction is increased if either (or, of course, both) has a high concentration. Again by conservation of mass we have the relations

$$\frac{d[NO_2]}{dt} = \frac{d[CO_2]}{dt} = +k_2[NO_3][CO].$$

Observe that the rate is always determined by the reacting chemicals.

For the reaction in which nitrogen dioxide decomposes into nitrogen monoxide and nitrogen trioxide,



we regard the left hand side as $NO_2 + NO_2$, so that the decay of nitrogen dioxide can be written,

$$\frac{d[NO_2]}{dt} = -2k_1[NO_2]^2.$$

Observe that the coefficient 2 is critical in this case and indicates that for each reaction that takes place, 2 molecules of nitrogen dioxide are used. The *exponent* 2 is a consequence of the law of mass action. By conservation of mass, we have

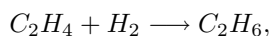
$$\frac{d[NO]}{dt} = \frac{d[NO_3]}{dt} = +k_1[NO_2]^2.$$

Finally, notice that the entire reaction $NO_2 + CO \longrightarrow NO + CO_2$ is modeled by a system of ODE,

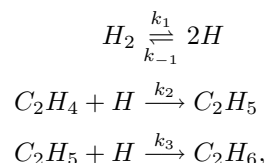
$$\begin{aligned} \frac{d[NO_3]}{dt} &= -k_2[NO_3][CO] + k_1[NO_2]^2 \\ \frac{d[CO]}{dt} &= -k_2[NO_3][CO] \\ \frac{d[NO_2]}{dt} &= -2k_1[NO_2]^2 + k_2[NO_3][CO]. \end{aligned}$$

Notice that we have a complete system of ODE and do not need to consider the concentrations of $[NO]$ and $[CO_2]$.

Example 2.3. In certain cases, a reaction can proceed in either direction. For example, in the hydrogenation of ethylene (C_2H_4) to ethane (C_2H_6),



a proposed mechanism is



where the first reaction can proceed in either direction. According to the law of mass action, we can model this mechanism with the following system of ODE,

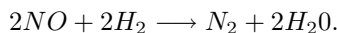
$$\begin{aligned} \frac{d[H_2]}{dt} &= -k_1[H_2] + k_{-1}[H]^2 \\ \frac{d[H]}{dt} &= 2k_1[H_2] - 2k_{-1}[H]^2 - k_2[C_2H_4][H] - k_3[C_2H_5][H] \\ \frac{d[C_2H_4]}{dt} &= -k_2[C_2H_4][H] \\ \frac{d[C_2H_5]}{dt} &= k_2[C_2H_4][H] - k_3[C_2H_5][H]. \end{aligned}$$

△

2.2.4 Determining Reaction Rates

In the examples above, we have assumed we know which reactions are elementary and which are not. A natural question becomes, how can we determine whether or not a given reaction is elementary?

Example 2.4. Consider the reaction between nitrogen monoxide and hydrogen, given by the stoichiometric equation,



Is this an elementary reaction?

In order to answer this question, we require some experimental data.² In Table 2.1, we record concentrations of nitrogen monoxide and hydrogen and initial reaction rates. Keep in mind that the units for concentrations are moles per unit volume, while the units for reaction rates are exactly the units of concentrations divided by time. I've left off the precise units in this example because I can't remember (and failed to write down) what the unit volume was.

Experiment	$[NO]$	$[H_2]$	Initial reaction rate $-\frac{d[NO]}{dt} = -\frac{d[H_2]}{dt}$
1	.1	.1	1.23×10^{-3}
2	.1	.2	2.46×10^{-3}
3	.2	.1	4.92×10^{-3}

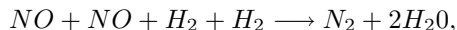
Table 2.1: Concentrations of nitrogen monoxide and hydrogen and initial reaction rates.

We now posit the general reaction rate $2k[NO]^a[H_2]^b$ for some three constants a , b , and k . That is, we assume

$$\frac{d[NO]}{dt} = \frac{d[H_2]}{dt} = -2k[NO]^a[H_2]^b,$$

²If you've taken first-year chemistry with a lab component you've almost certainly carried out an experiment similar to this one.

where the 2 is really just for show here since it could be subsumed into k . (Observe that if this reaction is elementary, we regard it as



for which a and b will both be 2.) For convenience of notation, we will write the positive rate $-\frac{d[NO]}{dt} = R$, so that we have

$$R = 2k[NO]^a[H_2]^b.$$

Taking now the natural logarithm of both sides, we find

$$\ln R = \ln 2k + a \ln[NO] + b \ln[H_2].$$

Given our data for R , $[NO]$, and $[H_2]$, we can use linear multivariate regression to determine the values of a , b , and k . That is, if we write $X = \ln[NO]$, $Y = \ln[H_2]$, and $Z = \ln R$, we have

$$Z = \ln 2k + aX + bY.$$

(In this case, we have exactly three pieces of data and three unknowns, so the fit will be precise, but in general we would have more data and we would proceed through regression.) In the following MATLAB code, N represents $[NO]$ and H represents H_2 .

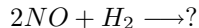
```
>>N=[.1 .1 .2];
>>H=[.1 .2 .1];
>>R=[1.23e-3 2.46e-3 4.92e-3];
>>M=[ones(size(N))' log(N)' log(H)'];
>>p=M\log(R)'
p =
0.2070
2.0000
1.0000
>>k=exp(.207)/2
k =
0.6150
```

In this case we determine that $a = 2$, $b = 1$, and $k = .615$, so that our rate law becomes

$$\frac{d[NO]}{dt} = -2 \cdot (.615)[NO]^2[H_2].$$

We conclude that this reaction is most likely not elementary.

So, what is the mechanism? Well—judging from our analysis, the first reaction might look something like



At this point, we need a chemist. △

2.2.5 Carbon Dating

Developed by the American physical chemist Willard F. Libby in 1947, carbon dating is a particular type of radioactive dating, applicable in cases for which the matter to be dated was once living. The radioactive isotope carbon-14 is produced at a relatively constant rate in the atmosphere, and like stable carbon-12, combines with oxygen to form carbon dioxide, which is incorporated into all living things. When an organism dies, its level of carbon-12 remains relatively constant, but its level of carbon-14 begins to decay with rate

$$\frac{d[C^{14}]}{dt} = -r[C^{14}], \quad r = 1.2097 \times 10^{-4} \text{ years}^{-1}$$

(this rate corresponds with the commonly quoted fact that carbon-14 has a half-life of 5730 years, by which we mean the level of carbon-14 in a substance is reduced by half after 5730 years). Since the fraction of

carbon-12 to carbon-14 remains relatively constant in living organisms (at the same level as it occurs in the atmosphere, roughly $[C^{14}]/[C^{12}] \cong 1.3 \times 10^{-12}$), we can determine how long an organism has been dead by measuring this ratio and determining how much carbon-14 has radiated away. For example, if we find that half the carbon-14 has radiated away, then we can say the material is roughly 5730 years old. In practice, researchers measure this ratio of carbon-14 to carbon-12 in units called *modern carbons*, in which the living ratio (ratio of carbon-14 to carbon-12 in a living organism) is defined to be 1 modern carbon.

Example 2.5. (Carbon dating the Shroud of Turin)³ The most famous (and controversial) case of carbon dating was that of the Shroud of Turin, which many believe to have covered Jesus of Nazareth in his tomb. In 1988, samples of the cloth were independently studied by three groups, one at the University of Arizona, one at Oxford University, and one at the Swiss Federal Institute of Technology (ETH) in Zurich. In this example, we will consider the data collected in Zurich. Five measurements were made on the level of modern carbon remaining in the shroud, .8755, .8766, .8811, .8855, and .8855 (two measurements led to the same number). Averaging these, we have a value $M = .8808$. Since the level of carbon-12 remains relatively constant, we can assume that the level of the ratio of carbon-14 to carbon-12 is reduced at the same rate as the level of carbon-14. We have, then

$$\frac{dM}{dt} = -rM; \quad r = 1.2097 \times 10^{-4} \text{ years}^{-1} \Rightarrow M(t) = M(0)e^{-rt}.$$

Setting $M(0) = 1$ as the level of modern carbon when the shroud was made, we need to find t so that

$$.8808 = e^{-1.2097 \times 10^{-4}t}.$$

Solving this relation, we find $t = 1049$ years, which dates the shroud to the year $1988 - 1049 = 939$ A.D. \triangle

2.3 Population Dynamics

In this section we will regard a population as just about any collection of objects we can count: animals, biological cells, automobiles etc. While the modeling of populations with differential equations is not precise (i.e., there is no equivalent to Newton's second law of motion), it can be extremely useful, and has aided especially in the areas of epidemic control and medical treatment. In this section, we will list and discuss most of the fundamental ODE models used in applications.

Throughout this section, we let $p(t)$ represent the total number of members of a population at time t .

1. Steady production. In the event that some population increases steadily (e.g., 5 members are created per hour), we have simply that the population's first time derivative is constant,

$$\frac{dp}{dt} = c; \quad p(0) = p_0 \Rightarrow p(t) = ct + p_0.$$

Examples include cars coming off an assembly line and T cells being created in bone marrow.

2. Malthusian model. Named for the British economist Thomas R. Malthus (1766–1834),⁴ the Malthusian model assumes that both the birth rate of a population and the death rate of a population are proportional to the current size of the population. For example, in a population of two people, the population will not

³The study we take our data from was originally published in *Nature* vol. 337 (1989), no. 6208 611–615, which is available at www.shroud.com/nature.htm. The data in the form I'm giving it here was given by Remi Van Haelst in his article *Radiocarbon Dating the Shroud of Turin: the Nature Report*, which is available at www.shroud.com/vanhels5.pdf. The results of this study have been widely disputed. One compelling argument against the date we find in this example is that the shroud was patched in medieval times and the examples studied were part of that patch.

⁴Malthus is perhaps the single most important figure in shaping current socioeconomic views of population dynamics. Prior to the publication of Malthus's *Essay on Population* in 1798, European statesmen and economists largely agreed that rising population was an indication of economic prosperity (this point of view was argued, for example, by the influential Scottish political economist and philosopher Adam Smith (1723–1790) in his *Wealth of Nations* (1776)). According to this point of view, if a king or ruling assemblage wanted to increase its nation's prosperity, it need only increase its pool of taxpayers. In his *Essay on Population*, Malthus pointed out that environments have finite resources, and consequently that rising populations must eventually lead to famine. Though increasingly clever methods of cultivation have allowed industrialized countries to sustain more people than Malthus would likely have thought possible, his thesis is now widely accepted.

grow very rapidly, but in a population of 6.2 billion people (roughly the earth’s population in 2004) growth is extremely rapid. Letting b represent birth rate and d represent death rate, we write,

$$\frac{dp}{dt} = bp - dp = rp; \quad p(0) = p_0 \Rightarrow p(t) = p_0 e^{rt},$$

where r , which is typically positive, will be referred to as the *growth rate* of the population.

3. Logistic model. A clear drawback of the Malthusian model is that it assumes there are no inherent limitations on the growth of a population. In practice, most populations have a size beyond which their environment can no longer sustain them. The logistic model incorporates this observation through the introduction of a “carrying capacity” K , the greatest population an environment can sustain. We have,

$$\frac{dp}{dt} = rp\left(1 - \frac{p}{K}\right); \quad p(0) = p_0 \Rightarrow p(t) = \frac{p_0 K}{(K - p_0)e^{-rt} + p_0}.$$

In order to better understand the role K plays, we recall the idea of equilibrium points or steady states (this will anticipate the general stability discussion of Section 3). An equilibrium point is some point at which a population quits changing: $\frac{dp}{dt} = 0$. In the case of the logistic equation, we can find all equilibrium points by solving the algebraic equation,

$$rp_e\left(1 - \frac{p_e}{K}\right) = 0 \Rightarrow p_e = 0, K.$$

We determine whether or not a population moves toward a particular equilibrium point by considering the sign of $\frac{dp}{dt}$ on either side of the equilibrium point. For the equilibrium point $p_e = K$, we observe that for $p > K$, $\frac{dp}{dt} < 0$ (that is, the population is decreasing), while for $p < K$, $\frac{dp}{dt} > 0$ (that is, the population is increasing). In this case, the population always approaches K , and we refer to K as a *stable* equilibrium point. *Very generally, stable equilibrium points represent long time behavior of solutions to ODE.*

4. Gompertz model. Named for the British actuary and mathematician Benjamin Gompertz (1779–1865), the Gompertz model is qualitatively similar to the logistic model. We have

$$\frac{dp}{dt} = -rp \ln\left(\frac{p}{K}\right); \quad p(0) = p_0.$$

The Gompertz model is often used in the study of tumor growth.

5. General single population model. The logistic and Gompertz models are both special cases of the general population model,

$$\frac{dp}{dt} = \frac{r}{a}p\left(1 - \left(\frac{p}{K}\right)^a\right),$$

where r and K play the same roles as in the logistic and Gompertz models, and a is typically fit to data. We note that $a = 1$ is the logistic model, and the Gompertz model is recovered from a limit as $a \rightarrow 0$.

6. Lotka-Volterra model. Named for the Italian mathematician Vito Volterra (1860–1940) and the Austrian chemist, demographer, ecologist, and mathematician Alfred J. Lotka (1880–1949), the Lotka–Volterra model describes the interaction between a predator (e.g., wildcats) with population $y(t)$ and its prey (e.g., rabbits) with population $x(t)$. We have,

$$\begin{aligned} \frac{dx}{dt} &= ax - bxy; & x(0) &= x_0 \\ \frac{dy}{dt} &= -ry + cxy; & y(0) &= y_0, \end{aligned}$$

where a , b , c , and r are all taken positive. We observe that in the absence of predators (i.e., in the case $y \equiv 0$) the prey thrive (they have Malthusian growth), while in the absence of prey (i.e., in the case $x \equiv 0$) the predators die off. The interaction or *predation* terms signify that the larger either the predator population or the prey population is, the more often the two populations interact, and that interactions tend to increase the predator population and decrease the prey population. While qualitatively enlightening, the Lotka–Volterra model isn’t robust enough to model many real interactions, though see Examples 2.6 and 2.7 in the course notes *Modeling Basics*.

7. Competition models. In addition to predator–prey interactions, we would often like to model two species such as rabbits and deer that compete for the same resources. In the (unlikely) event that each species uses exactly the same amount of the environment, we could model this interaction through the ODE,

$$\begin{aligned}\frac{dx}{dt} &= r_1x\left(1 - \frac{x+y}{K}\right) \\ \frac{dy}{dt} &= r_2y\left(1 - \frac{x+y}{K}\right),\end{aligned}$$

where we have simply asserted that if the total population $x + y$ exceeds the carrying capacity, both populations will begin to die off. More generally, we assume that each population has a different carrying capacity and a different interaction with its environment, and only keep this general idea that if either population gets sufficiently large, the other will begin to die off. Under this assumption, a reasonable model is,

$$\begin{aligned}\frac{dx}{dt} &= r_1x\left(1 - \frac{x}{K_1} - \frac{s_1y}{K_1}\right) \\ \frac{dy}{dt} &= r_2y\left(1 - \frac{y}{K_2} - \frac{s_2x}{K_2}\right),\end{aligned}$$

where K_1 represents the carrying capacity of species x , K_2 represents the carrying capacity of species y , and s_1 represents a scaling for the amount of species x 's environment used by species y (and similarly for s_2). For example, suppose species y is larger and eats roughly twice as much per animal as species x . Then we take $s_1 = 2$. It seems fairly natural that if $s_1 = 2$, then $s_2 = 1/2$. That is, if species y uses twice the environment of species x , then species x uses half the environment of species y . While intuitively satisfying, this reciprocity doesn't always fit the data.

8. The SIR epidemic model The most simple model for studying the spread of epidemics involves three populations, the *susceptible* members, $S(t)$, the *infected* members, $I(t)$, and the *removed* members, $R(t)$. (The removed members can either have recovered (in which case they are assumed in this model to be immune) or died.) The SIR model takes the form,

$$\begin{aligned}\frac{dS}{dt} &= -aSI \\ \frac{dI}{dt} &= aSI - bI \\ \frac{dR}{dt} &= bI.\end{aligned}$$

In the first equation, we observe that the rate at which susceptible members of the population become infected is proportional to the number of interactions there are between members of the population. The second equation records that each member lost from $S(t)$ moves to population $I(t)$ and that members of $I(t)$ recover or die at some rate b determined by the disease and typically found experimentally.

9. Half saturation constants. In the Lotka–Volterra predator–prey model above, the predator growth due to predation takes the form $+cxy$. Even if there is only one predator left, this claims that if there are enough prey, the predators will continue to grow rapidly. A better expression might be,

$$\frac{cxy}{x+M},$$

for which there is an intrinsic limit on how fast the predator population can grow when saturated with prey. In particular, at full saturation, we consider the limit as the prey population goes to infinity,

$$\lim_{x \rightarrow \infty} \frac{cxy}{x+M} = cy,$$

which is the full saturation growth. We refer to the constant M as the “half saturation” constant, because when $x = M$, the growth rate is at precisely half saturation,

$$\frac{cxy}{x+M} = \frac{1}{2}cy.$$

10. Learning terms. We often want to specify in our model that a species becomes more (or less) adept at some function as time progresses. For example, we might find in a predator–prey situation that prey learn over time to avoid predators. In this case, the growth and decay rates due to predation will depend on the independent variable t . Typically, we assume this change is slow, and logarithmic terms are often employed. In the Lotka–Volterra model, under the assumption that the prey learn to avoid the predators (and the predators do not get more adept at finding the prey), we could write,

$$\begin{aligned}\frac{dx}{dt} &= ax - \frac{b}{\ln(e+t)^k}xy \\ \frac{dy}{dt} &= -ry + \frac{c}{\ln(e+t)^k}xy,\end{aligned}$$

where we evaluate natural log at $e+t$ so that we get 1 when $t=0$, and k is a new parameter to be fit to data.

11. Delay models. One of the deficiencies with the population models discussed above is that in each case birth rate is assumed to change instantly with a change in population. More generally, we might expect the members of a population to reach some threshold age before giving birth, introducing a time delay into the model. For example, a time-delay Malthusian model would take the form

$$\frac{dp}{dt} = rp(t-T),$$

wherein the growth rate of the population at time t depends on the population at time $t-T$.

12. Discrete time models.⁵ In a discrete time population model, we increment time by some discrete amount and update our population according to some rule. One frequently used discrete model, similar to the continuous logistic model, takes the form

$$p_{t+1} = p_t e^{r(1-\frac{p_t}{K})}.$$

13. Matrix models. Another type of discrete population model involves the notion of a *transition* matrix. In this case, we consider a number of different categories the population can be classified in, and study the probability of a member from one category of the population moving to another category. For example, a recent paper in the journal *Conservation Biology* [P. C. Cross and S. R. Beissinger, *Using logistic regression to analyze the sensitivity of PVA models: a comparison of models based on African wild dog model*, *Conservation Biology* **15** (2001), no. 5, 1335–1346.] considers African wild dogs, dividing the population into three categories: pups, yearlings and adults. We assume pups survive to become yearlings with probability S_p , that yearlings reproduce pups with probability R_y , that yearlings survive to adulthood with probability S_y , that adults reproduce pups with probability R_a , and finally that adults have an annual probability of survival S_a (see Figure 2.2).

Suppose we know initial populations P_0 , Y_0 , and A_0 , and would like to determine the corresponding populations a year later. For pups, we have loss as they become yearlings and gain as the yearlings and adults reproduce. Arguing similarly for yearlings and adults, we arrive at the model,

$$\begin{aligned}P_1 &= R_y Y_0 + R_a A_0 \\ Y_1 &= + S_p P_0 \\ A_1 &= + S_y Y_0 + S_a A_0.\end{aligned}$$

(It might be tempting to think we should have a loss term in the pups equation of the form $-S_p P_0$, but keep in mind that after one year *all* pups are lost (if they’ve survived, they’ve become yearlings), and we have only gotten new pups from reproduction.) Of course, we can write this last expression in the matrix form,

$$\begin{pmatrix} P_1 \\ Y_1 \\ A_1 \end{pmatrix} = \begin{pmatrix} 0 & R_y & R_a \\ S_p & 0 & 0 \\ 0 & S_y & S_a \end{pmatrix} \begin{pmatrix} P_0 \\ Y_0 \\ A_0 \end{pmatrix}.$$

⁵The final two models in this section are not ODE models, but are well worth mentioning.

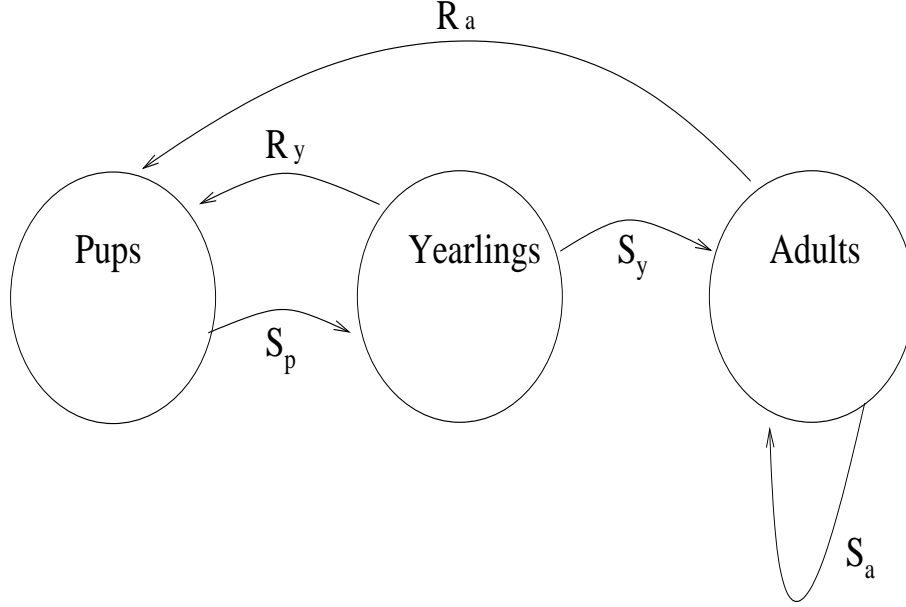


Figure 2.2: Transitions in African wild dog population categories.

Similarly, given the year 1 populations, we can produce the year 2 populations through,

$$\begin{pmatrix} P_2 \\ Y_2 \\ A_2 \end{pmatrix} = \begin{pmatrix} 0 & R_y & R_a \\ S_p & 0 & 0 \\ 0 & S_y & S_a \end{pmatrix} \begin{pmatrix} P_1 \\ Y_1 \\ A_1 \end{pmatrix} = \begin{pmatrix} 0 & R_y & R_a \\ S_p & 0 & 0 \\ 0 & S_y & S_a \end{pmatrix}^2 \begin{pmatrix} P_0 \\ Y_0 \\ A_0 \end{pmatrix}.$$

More generally, the year k population can now be determined as

$$\begin{pmatrix} P_k \\ Y_k \\ A_k \end{pmatrix} = \begin{pmatrix} 0 & R_y & R_a \\ S_p & 0 & 0 \\ 0 & S_y & S_a \end{pmatrix}^k \begin{pmatrix} P_0 \\ Y_0 \\ A_0 \end{pmatrix}.$$

The matrix in these calculations is typically referred to as the transition matrix.

2.4 Newtonian mechanics.

Given a point particle with position $x(t)$ with constant mass, Newton's second law of motion, $F = ma$, can be written as an ODE,

$$F = ma = m \frac{dv}{dt} = m \frac{d^2x}{dt^2}.$$

In the event that the mass of our particle is changing, we have the generalized form,

$$F = \frac{dp}{dt},$$

where $p = mv$ is momentum.

Example 2.6. (Drag racing) Recall that for an object on a flat surface, the force due to friction is proportional to the object's weight, with proportionality constant the coefficient of static or dynamic friction,

$$F = -\mu mg.$$

Observe in this relationship that we can reasonably regard $\mu \leq 1$. If not, then it would take less force to lift the object and carry it than it would to push it. Since the entire force pushing a dragster forward is due to

friction (between the tires and the road), we expect the maximum force propelling the dragster forward to be $F = mg$. Under this assumption, we can determine the minimum time it will take a dragster to complete a standard quarter-mile course (402.34 meters). If $x(t)$ represents position at time t along the course (with initial position and initial velocity assumed 0), then we have, according to Newton's second law,

$$\frac{d^2x}{dt^2} = g \Rightarrow x(t) = \frac{1}{2}gt^2.$$

We compute the minimum track time as,

$$t = \sqrt{\frac{2(402.34)}{9.81}} = 9.06 \text{ seconds.}$$

Let's put this to the test. On June 2, 2001, Kenny Bernstein set the world record for a quarter mile track with a time $t = 4.477$ seconds.⁶ △

Example 2.7. (Planetary motion) Consider the earth–sun system in two space dimensions. We choose some arbitrary origin $(0, 0)$ and let $r_1 = (x_1, y_1)$ represent the position of the sun (mass M) relative to the origin and $r_2 = (x_2, y_2)$ represent the position of the earth (mass m) relative to this origin. (See Figure 2.3.)

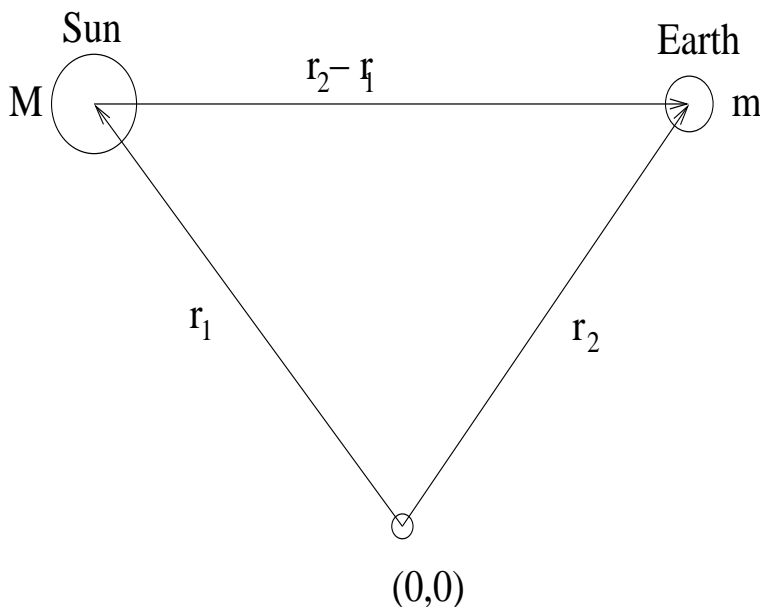


Figure 2.3: The earth–sun system in two dimensions.

According to Newton's law of gravitation, the magnitude of the force exerted by one (point) mass on another is proportional to the product of the masses and inversely proportional to the distance between the masses squared, with constant of proportionality G . Ignoring direction, we have

$$F = \frac{GMm}{d^2}.$$

In order to incorporate direction, we assume the force on either mass is directed radially toward the other mass. The force on the sun due to the earth is given by,

$$F_{\text{sun}} = \frac{GMm}{|r_2 - r_1|^3}(r_2 - r_1),$$

⁶Race car drivers “burn out” their tires at the beginning of a race, and this makes the tires adhere to the racing surface, so that they can “push off.” Viewed another way, the cars get more difficult to lift.

while the force on the earth due to the sun is given by,

$$F_{\text{earth}} = -\frac{GMm}{|r_2 - r_1|^3}(r_2 - r_1).$$

Finally, according to Newton's second law of motion, we can set $F = ma$, for which we obtain the vector ODE

$$\begin{aligned} M \frac{d^2 r_1}{dt^2} &= \frac{GMm}{|r_2 - r_1|^3}(r_2 - r_1) \\ m \frac{d^2 r_2}{dt^2} &= -\frac{GMm}{|r_2 - r_1|^3}(r_2 - r_1), \end{aligned}$$

or component-wise

$$\begin{aligned} x_1'' &= \frac{Gm(x_2 - x_1)}{((x_2 - x_1)^2 + (y_2 - y_1)^2)^{3/2}}, \\ y_1'' &= \frac{Gm(y_2 - y_1)}{((x_2 - x_1)^2 + (y_2 - y_1)^2)^{3/2}}, \\ x_2'' &= -\frac{GM(x_2 - x_1)}{((x_2 - x_1)^2 + (y_2 - y_1)^2)^{3/2}}, \\ y_2'' &= -\frac{GM(y_2 - y_1)}{((x_2 - x_1)^2 + (y_2 - y_1)^2)^{3/2}}. \end{aligned}$$

△

2.4.1 Newtonian Mechanics in Polar Coordinates

In Example 2.7, we described planetary motion in the standard cartesian coordinate system, for which our unit vectors are $e_x = (1, 0)$ and $e_y = (0, 1)$. For many physical systems, we find it convenient to describe motion in polar coordinates, for which a point P is described by its distance from the origin r and its angle from the horizontal axis θ . In this case, the unit vectors are e_r and e_θ , where e_r is a vector of unit length directed from the origin to the point P and e_θ is a vector of unit length orthogonal to e_r (see Figure 2.4).

In the event of an incremental change in angle θ , $d\theta$, we have incremental changes in e_r and e_θ , described by

$$de_r \cong e_\theta d\theta; \quad de_\theta \cong -e_r d\theta,$$

where the approximation approaches equality as $d\theta \rightarrow 0$. This is perhaps easiest to see by observing from Figure 2.4 first that these expressions have the correct direction and that in each case the length of the incremental vector approximately the arclength $1 \cdot d\theta$. We conclude the critical relations,

$$\frac{de_r}{d\theta} = e_\theta; \quad \frac{de_\theta}{d\theta} = -e_r.$$

We are now in a position to develop an expression for acceleration in polar coordinates. Let $\vec{x}(t)$ be the position vector of some point particle at time t . Then \vec{x} can be described in polar coordinates as $\vec{x}(t) = r(t)e_r(t)$, where we keep in mind that e_r is a vector. That is, $e_r(t)$ will always be a unit vector pointing in the direction of $\vec{x}(t)$ and r will be a scalar equal in value to the length of $\vec{x}(t)$. We compute

$$\frac{d\vec{x}}{dt} = r_t e_r + r \frac{de_r}{dt} = r_t e_r + r e_\theta \frac{d\theta}{dt},$$

and similarly

$$\frac{d^2 \vec{x}}{dt^2} = (r_{tt} - r(\theta_t)^2)e_r + (2r_t \theta_t + r\theta_{tt})e_\theta.$$

That is, acceleration in the radial direction is given by $r_{tt} - r(\theta_t)^2$, while acceleration in the angular direction is given by $2r_t \theta_t + r\theta_{tt}$. In the case of a *central* force such as gravity, all force is in the radial direction, and

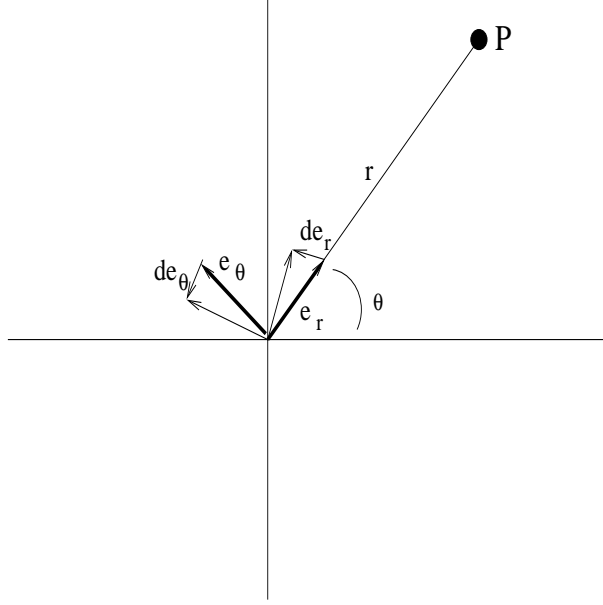


Figure 2.4: Unit vectors in polar coordinates.

Newton's second law of motion becomes the system

$$\begin{aligned} m(r_{tt} - r(\theta_t)^2) &= -F \\ m(2r_t\theta_t + r\theta_{tt}) &= 0. \end{aligned}$$

In particular, if we assume the sun's position is fixed in space, the earth's motion around it can be described by the system

$$\begin{aligned} r_{tt} - r(\theta_t)^2 &= -\frac{GM}{r^2} \\ 2r_t\theta_t + r\theta_{tt} &= 0. \end{aligned}$$

2.5 Hamiltonian Mechanics

Consider an object of mass m with height $y(t)$ falling under the influence of gravity only. We have

$$\begin{aligned} \text{Kinetic Energy} &= \frac{1}{2}mv^2 = \frac{p^2}{2m} \\ \text{Potential Energy} &= mgy, \end{aligned}$$

where p is the system momentum and *potential energy* is described through the function $U(y)$ such that

$$\text{Force} = -\frac{\partial U}{\partial y}.$$

The *Hamiltonian*, defined by

$$H(p, y) = \text{Kinetic Energy} + \text{Potential Energy} = \frac{p^2}{2m} + mgy$$

represents the entire energy of the system. For a conservative system (in the event that energy is conserved) we have $\frac{dH}{dt} = 0$, from which we find

$$\frac{dH}{dt} = \frac{\partial H}{\partial p} \frac{dp}{dt} + \frac{\partial H}{\partial y} \frac{dy}{dt} = 0. \quad (2.3)$$

Since kinetic energy is independent of position ($\frac{\partial K.E.}{\partial y} = 0$), we must have

$$\frac{\partial H}{\partial y} = \frac{\partial U}{\partial y} = -F.$$

In this way, Newton's second law can be written as $-\frac{\partial H}{\partial y} = \frac{dp}{dt}$, and (2.3) becomes $\frac{\partial H}{\partial p} = \frac{dy}{dt}$. We have, then, for conservative systems,

$$\begin{aligned}\frac{dp}{dt} &= -\frac{\partial H}{\partial y} \\ \frac{dy}{dt} &= \frac{\partial H}{\partial p},\end{aligned}$$

by which we see that if we can write down the total energy (Hamiltonian) for a conservative system, we can write down its equation of motion.

2.6 Variational Methods

Consider a point (x_1, y_1) on a curve emerging from the origin. The arclength of such a curve is given by the *functional*

$$F[y] = \int_0^{x_1} \sqrt{1 + y'(x)^2} dx,$$

where by *functional* we mean a mapping that takes functions as its input and returns numbers as output. Suppose we want to determine the shortest possible path between $(0, 0)$ and (x_1, y_1) . (Intuitively, of course, we know this is a line.) The question becomes: What curve $y(x)$ minimizes the functional F . What we would like to do, of course, is take a derivative of F and set it to 0. The difficulty lies in taking a derivative with respect to a function (the input for F). The study of such a theory is referred to as the *calculus of variations*.

In order to set some notation, we will designate the minimizing function as $y_e(x)$ and define a family of *variations* of this minimizer as the functions

$$y_s(x) = y_e(x) + sy(x),$$

where the functions $y(x)$ can be *any* continuously differentiable functions for which $y(0) = y(x_1) = 0$ (we assume $y_s(x)$ and $y_e(x)$ agree at the endpoints). We say that $y(x)$ belongs to the *function class* $C_0^1[0, x_1]$: the collection of all continuously differentiable functions on $x \in [0, x_1]$ that vanish at the endpoints.

According to our assumption, $F[y_e] = \text{minimum}$, and consequently

$$F[y_s] \Big|_{s=0} = \min \Rightarrow \frac{\partial}{\partial s} F[y_s] \Big|_{s=0} = 0.$$

That is, since $\phi(s) := F[y_s]$ is minimized at $s = 0$ (for *any* $y(x)$), its s -derivative at $s = 0$ must be 0. We have

$$F[y_s] = F[y_e + sy] = \int_0^{x_1} \sqrt{1 + (y'_e + sy')^2} dx,$$

from which

$$\frac{\partial}{\partial s} F[y_s] = \int_0^{x_1} \frac{(y'_e + sy')y'}{\sqrt{1 + (y'_e + sy')^2}} dx.$$

Upon setting $\frac{\partial}{\partial s} F[y_s] \Big|_{s=0} = 0$, we have

$$\int_0^{x_1} \frac{y'_e y'}{\sqrt{1 + (y'_e)^2}} dx = 0.$$

Integrating by parts, we find that this integral equation can be re-written as

$$\int_0^{x_1} \frac{y''_e y}{(1 + (y'_e)^2)^{3/2}} dx = 0.$$

At this point, we argue as follows: since $y(x)$ can be almost any function we choose (it only needs to be continuously differentiable and to vanish on the boundary), we can choose it to always have the same sign as y_e'' . In this case, the numerator and denominator of our integrand will both necessarily be positive, and consequently the only chance our integral has of being 0 is if $y_e'' = 0$. In that case, we have the boundary value problem

$$\begin{aligned} y_e'' &= 0 \\ y(0) &= 0 \\ y(x_1) &= y_1. \end{aligned}$$

In this way we have converted the problem of minimizing a functional into the problem of solving an ODE.

3 Well-posedness Theory

Despite the clunky name, *well-posedness* analysis is one of the most important things for an applied mathematician to understand. In order to get an idea of the issues involved, we will consider the example of a pendulum, initially perturbed but otherwise under the influence of gravity alone.

Example 3.1. Consider the motion of a mass, m , swinging at the end of a rigid rod, as depicted in Figure 3.1. Assume air resistance is negligible.

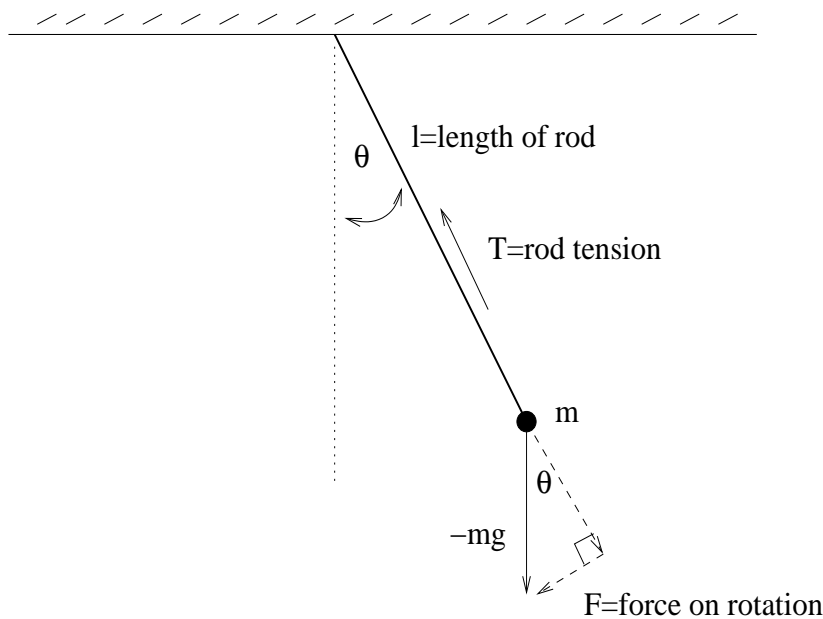


Figure 3.1: Pendulum motion under the influence of gravity alone.

The force due to gravity on m acts vertically downward, and must be decomposed into a force $-T$, which is exactly balanced by the rod, and a force F , directed tangentially to the arc of motion. Observing the right triangle, with hypotenuse of length $-mg$, we have

$$\begin{aligned} \cos \theta &= -\frac{T}{mg} \Rightarrow T = -mg \cos \theta, \\ \sin \theta &= -\frac{F}{mg} \Rightarrow F = -mg \sin \theta. \end{aligned}$$

Measuring distance as arclength, $d = l\theta$, Newton's second law of motion ($F = ma$) determines

$$\begin{aligned} \frac{d^2\theta}{dt^2} &= -\frac{g}{l}\sin\theta \\ \theta(0) &= \theta_0, \quad \frac{d}{dt}\theta(0) = \omega_0. \end{aligned} \tag{3.1}$$

In order to solve equation (3.1) with MATLAB, we must first write it as a first order system. Taking $x_1 = \theta$ and $x_2 = \frac{d\theta}{dt}$, we have

$$\begin{aligned} \frac{dx_1}{dt} &= x_2; \quad x_1(0) = \theta_0, \\ \frac{dx_2}{dt} &= -\frac{g}{l}\sin x_1; \quad x_2(0) = \omega_0. \end{aligned} \tag{3.2}$$

Taking $l = 1$, we will store this equation in the MATLAB M-file *pendode.m*,

```
function xprime = pendode(t,x);
%PENDODE: Holds ODE for pendulum equation.
g = 9.81; l = 1;
xprime = [x(2); -(g/l)*sin(x(1))];
```

and solve it with the M-file *pend.m*,

```
function f = pend(theta0,v0);
%PEND: Solves and plots ODE for pendulum equation
%Inputs are initial angle and initial angular velocity
x0 = [theta0 v0];
tspan = [0 5];
[t,x] = ode45(@pendode,tspan,x0);
plot(x(:,1),x(:,2));
```

Taking initial angle $\pi/4$ and initial velocity 0 with the command *pend(pi/4,0)*, leads to Figure 3.2 (I've added the labels from MATLAB's pop-up graphics window).

Notice that time has been suppressed and the two dependent variables x_1 and x_2 have been plotted in what we refer to as a *phase portrait*. Beginning at the initial point $\theta_0 = \frac{\pi}{4}$, $\omega_0 = 0$ (the right-hand tip of the football), we observe that angular velocity becomes negative (the pendulum swings to the left) and angle decreases. At the bottom of the arc, the angle is 0 but the angular velocity is at a maximum magnitude (though negatively directed), while at the left-hand tip of the football the object has stopped swinging (instantaneously), and is turning around. The remainder of the curve corresponds with the object's swinging back to its starting position. In the (assumed) absence of air resistance or other forces, the object continues to swing like this indefinitely. Alternatively, taking initial angle 0 and initial velocity 10 with the command *pend(0,10)* leads to Figure 3.3.

Observe that in this case angular velocity is always positive, indicating that the pendulum is always swinging in the same (angular) direction: we have started it with such a large initial velocity that it's looping its axis.

Now that we have a fairly good idea of how to understand the pendulum phase diagrams, we turn to the critical case in which the pendulum starts pointed vertically upward from its axis (remember that we have assumed it is attached to a rigid rod). After changing the variable *tspan* in *pend* to $[0, 20]$ (solving now for 20 seconds), the command *pend(pi,0)* leads to Figure 3.4. In the absence of any force other than gravity, we expect our model to predict that the pendulum remains standing vertically upward. (What could possibly cause it to fall one way rather than the other?) What we find, however, is that our model predicts that it will fall to the left and then begin swinging around its axis.

Consider finally a change in this last initial data of one over one trillion ($10^{-12} = .000000000001$). The MATLAB command *pend(pi+1e-12,0)* produces Figure 3.5. We see that with a change in initial data as small as 10^{-12} radians, the change in behavior is enormous: the pendulum spins in the opposite direction. We conclude that our model, at least as it is solved on MATLAB, fails at the initial data point $(\pi, 0)$. In particular, we say that our model is not well-posed at this point. \triangle

In general, for well-posedness, we will require three things of a model:

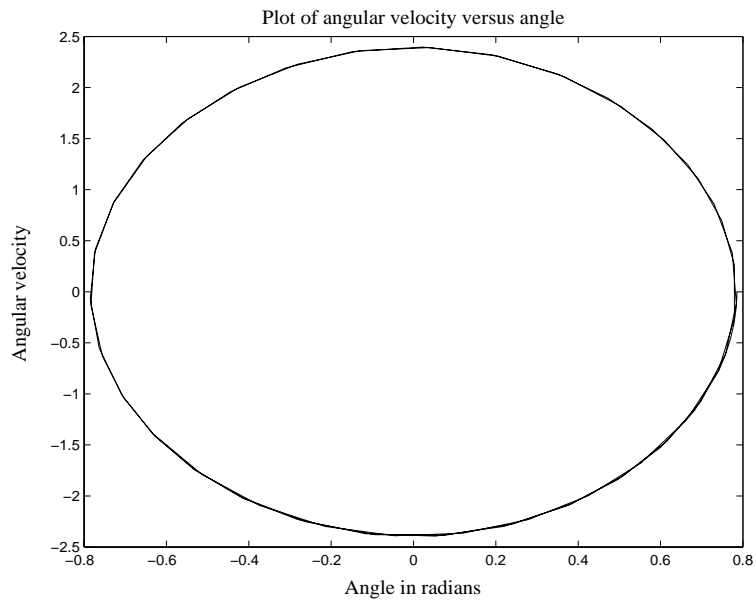


Figure 3.2: Pendulum motion for the case $\theta_0 = \frac{\pi}{4}$ and $\omega_0 = 0$.

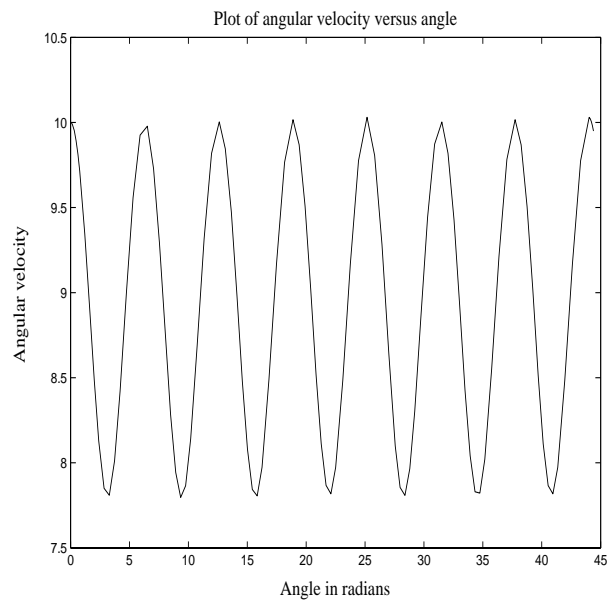


Figure 3.3: Pendulum motion for the case $\theta_0 = 0$ and $\omega_0 = 10 \text{ s}^{-1}$.

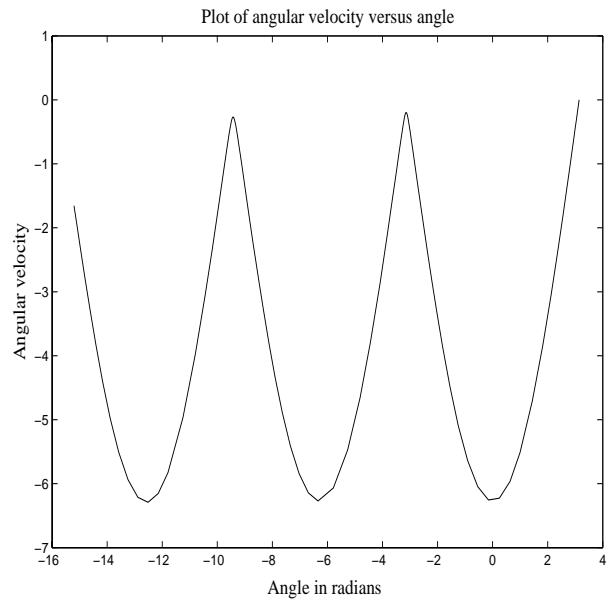


Figure 3.4: Pendulum motion for the case $\theta_0 = \pi$ and $\omega_0 = 0 \text{ s}^{-1}$.

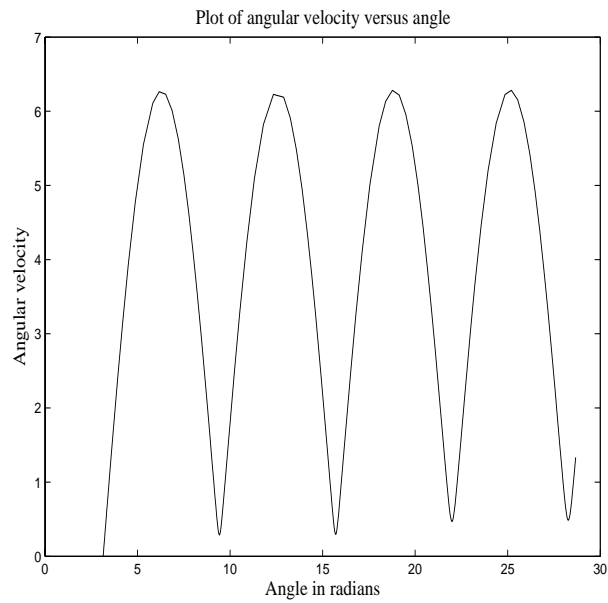


Figure 3.5: Pendulum motion for the case $\theta_0 = \pi + 10^{-12}$ and $\omega_0 = 0 \text{ s}^{-1}$.

1. (Existence) There exists a solution to the model.
2. (Uniqueness) The solution is unique.
3. (Stability) The solution does not change dramatically if we only change the initial data a little.

In the next three sections, we will consider each of these in turn, beginning with stability and working our way back to the most abstract theory, existence.

3.1 Stability Theory

The difficulty we ran into in Example 4.1 is with stability. Near the initial data point $(\pi, 0)$, small changes in initial data lead to dramatic changes in pendulum behavior.

Example 3.1 continued. For systems of two first-order differential equations such as (3.2), we can study phase diagrams through the useful trick of dividing one equation by the other. We write,

$$\frac{dx_2}{dx_1} = \frac{\frac{dx_2}{dt}}{\frac{dx_1}{dt}} = \frac{-\frac{g}{l} \sin x_1}{x_2},$$

(the *phase-plane equation*) which can readily be solved by the method of separation of variables for solution

$$\frac{x_2^2}{2} = \frac{g}{l} \cos x_1 + C. \tag{3.3}$$

At $t = 0$, $x_1(0) = \theta_0$ and $x_2(0) = \omega_0$, fixing C . We will create a phase plane diagram with the M-file *penphase.m*.

```
function f = penphase(theta,w0);
%PENPHASE: Plots phase diagram for
%pendulum equation with initial angle theta
%and initial angular velocity w0.
g = 9.81; l = 1.0;
C = w0^2/2 - (g/l)*cos(theta);
if C > g/l
x = linspace(-pi,pi,50);
else
maxtheta = acos(-C*l/g); %Maximum value of theta
x = linspace(-maxtheta,maxtheta,50);
end
up = sqrt(2*g/l*cos(x)+2*C);
down = -sqrt(2*g/l*cos(x)+2*C);
plot(x,up);
hold on
plot(x,down);
```

Typing in sequence *penphase(pi/12,0)*, *penphase(pi/4,0)*, *penphase(pi/2,0)*, *penphase(pi,0)*, *penphase(pi/4,6)*, we create the phase plane diagram given in Figure 3.6.

The point $(\theta_0, \omega_0) = (0, 0)$ corresponds with the pendulum's hanging straight down, while the points $(\theta_0, \omega_0) = (\pi, 0)$ and $(\theta_0, \omega_0) = (-\pi, 0)$ both correspond with the pendulum's standing straight up above its axis. Notice that at each of these *critical* or *equilibrium* points our model analytically predicts that the pendulum will not move. For example, at $(\theta_0, \omega_0) = (0, 0)$ we find from (3.2) that $\frac{dx_1}{dt} = \frac{dx_2}{dt} = 0$: the angle and angular velocity are both zero, so the pendulum remains at rest.

Definition. (Equilibrium point) For an autonomous system of ordinary differential equations

$$\frac{dx}{dt} = f(x), \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix},$$

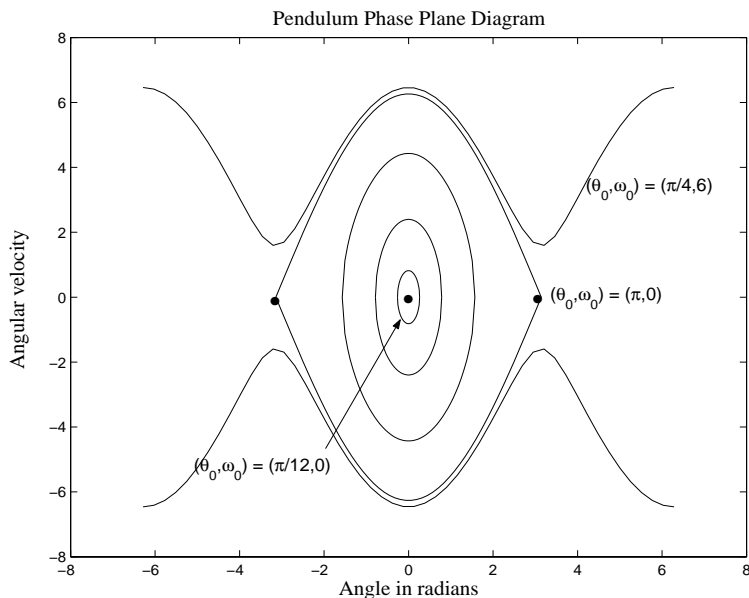


Figure 3.6: Phase plane diagram for a simple pendulum (Example 3.1 continued).

we refer to any point (x_0) so that $f(x_0) = 0$ as an *equilibrium point*.

Typically, equilibrium points govern long time behavior of physical models. In particular, solutions tend to approach stable equilibrium points as time gets large, and to move away from unstable equilibrium points. Perturbing the initial point $(0, 0)$ a little (pushing the pendulum slightly to the right or left), we observe that the pendulum's behavior changes only slightly: if we push it one millimeter to the right, it will swing back and forth with maximum displacement one millimeter. On the other hand, as we have seen, if we perturb the initial point $(\pi, 0)$ the pendulum's behavior changes dramatically. We say that $(0, 0)$ is a stable equilibrium point and that $(\pi, 0)$ and $(-\pi, 0)$ are both unstable equilibrium points. More precisely, we say that the point $(0, 0)$ is *orbitally stable*, which signifies that solutions don't actually approach it, but rather remain near it.

In general, we can study stability without solving equations quite as complicated as (3.3). Suppose we want to analyze stability at the point $(0, 0)$. We first recall the Taylor expansion of $\sin x$ near $x = 0$,

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots$$

For x near 0, higher powers of x are dominated by x , and we can take the approximation, $\sin x \cong x$, which leads to the *linearized* equations,

$$\begin{aligned} \frac{dx_1}{dt} &= x_2 \\ \frac{dx_2}{dt} &= -\frac{g}{l}x_1. \end{aligned} \tag{3.4}$$

(That is, the right-hand sides of (3.4) are both linear, which will always be the case when we take the linear terms from a Taylor expansion about an equilibrium point.) Developing the phase plane equation as before, we now have

$$\frac{dx_2}{dx_1} = \frac{\frac{dx_2}{dt}}{\frac{dx_1}{dt}} = \frac{-\frac{g}{l}x_1}{x_2},$$

with solution

$$\frac{x_2^2}{2} + \frac{g}{l} \cdot \frac{x_1^2}{2} = C,$$

which corresponds with ellipses centered at $(0, 0)$ with radial axis lengths $\sqrt{2C}$ and $\sqrt{2lC/g}$ (see Figure 3.7). Typically such solutions are referred to as *integral curves*. Returning to equations (3.4), we add direction along the ellipses by observing from the first equation that for $x_2 > 0$, x_1 is increasing, and for $x_2 < 0$, x_1 is decreasing. The directed sections of integral curves along which the object moves are called *trajectories*. Our stability conclusion is exactly the same as we drew from the more complicated Figure 3.6. In particular, in the case that we have closed loops about an equilibrium point, we say the point is *orbitally stable*.

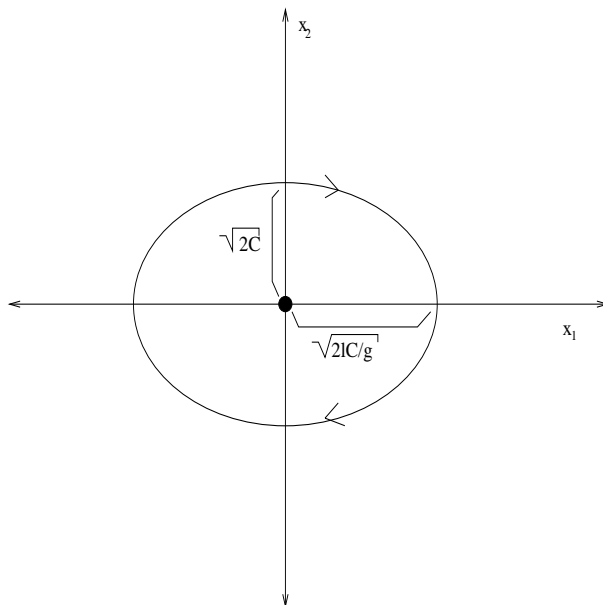


Figure 3.7: Phase plane diagram near the equilibrium point $(0, 0)$.

For the point $(\pi, 0)$ we first make the change of variables,

$$\begin{aligned}x_1 &= \pi + y_1 \\x_2 &= 0 + y_2,\end{aligned}$$

and observe that in the variables y_1 and y_2 the equilibrium point is again at $(0, 0)$. In these variables, our system becomes,

$$\begin{aligned}\frac{dy_1}{dt} &= y_2 \\ \frac{dy_2}{dt} &= -\frac{g}{l} \sin(\pi + y_1).\end{aligned}$$

Recalling the Taylor expansion of $\sin y_1$ at the point π ,

$$\sin(\pi + y_1) = \sin \pi + (\cos \pi)y_1 - \frac{\sin \pi}{2}y_1^2 + \dots,$$

we arrive at the new linearized equation,

$$\begin{aligned}\frac{dy_1}{dt} &= y_2 \\ \frac{dy_2}{dt} &= \frac{g}{l}y_1.\end{aligned}$$

Proceeding exactly as above we again write the phase plane equation,

$$\frac{dy_2}{dy_1} = \frac{\frac{dy_2}{dt}}{\frac{dy_1}{dt}} = \frac{\frac{g}{l}y_1}{y_2},$$

which can be solved by the method of separation of variables for implicit solution,

$$-\frac{y_2^2}{2} + \frac{g}{l} \frac{y_1^2}{2} = C,$$

which corresponds with hyperbolas (see Figure 3.8). Observe that in this case all trajectories move first toward the equilibrium point and then away. We refer to such an equilibrium point as an *unstable saddle*. \triangle

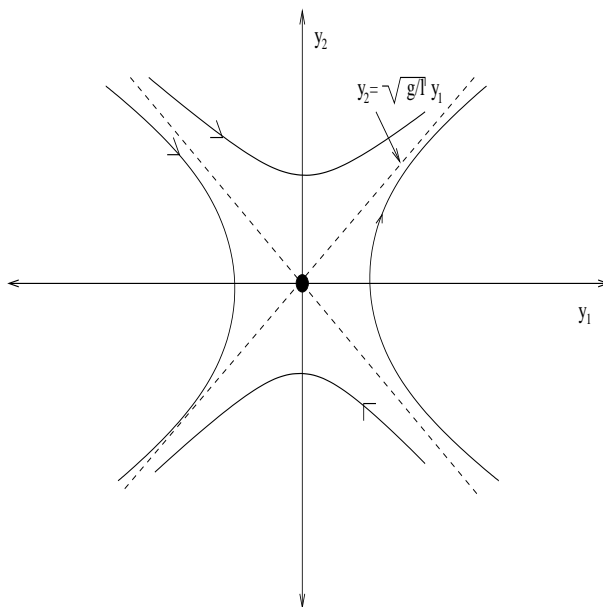


Figure 3.8: Phase plane diagram near the equilibrium point $(\pi, 0)$.

Example 3.2. As a second example of stability analysis, we will consider the Lotka–Volterra predator–prey equations,

$$\begin{aligned} \frac{dx}{dt} &= ax - bxy \\ \frac{dy}{dt} &= -ry + cxy. \end{aligned} \tag{3.5}$$

First, we find all equilibrium points by solving the system of algebraic equations,

$$\begin{aligned} ax - bxy &= 0 \\ -ry + cxy &= 0. \end{aligned}$$

We find two solutions, $(x_1, y_1) = (0, 0)$ and $(x_2, y_2) = (\frac{r}{c}, \frac{a}{b})$. The first of these corresponds with an absence of both predator and prey, and of course nothing happens (in the short term). The second is more interesting, a point at which the predator population and the prey population live together without either one changing. If this second point is unstable then any small fluctuation in either species will destroy the equilibrium and one of the populations will change dramatically. If this second point is stable then small fluctuations in species population will not destroy the equilibrium, and we would expect to observe such equilibria in nature. In this way, *stability typically determines physically viable behavior*.

In order to study the stability of this second point, we first linearize our equations by making the substitutions

$$\begin{aligned} x &= \frac{r}{c} + z_1 \\ y &= \frac{a}{b} + z_2. \end{aligned}$$

Substituting x and y directly into equation (3.5) we find

$$\begin{aligned}\frac{dz_1}{dt} &= a\left(\frac{r}{c} + z_1\right) - b\left(\frac{r}{c} + z_1\right)\left(\frac{a}{b} + z_2\right) = -\frac{br}{c}z_2 - bz_1z_2 \\ \frac{dz_2}{dt} &= -r\left(\frac{a}{b} + z_2\right) + c\left(\frac{r}{c} + z_1\right)\left(\frac{a}{b} + z_2\right) = \frac{ca}{b}z_1 + cz_1z_2.\end{aligned}$$

(Observe that in the case of polynomials a Taylor expansion emerges from the algebra, saving us a step.) Dropping the nonlinear terms, we arrive at our linear equations,

$$\begin{aligned}\frac{dz_1}{dt} &= -\frac{br}{c}z_2 \\ \frac{dz_2}{dt} &= \frac{ca}{b}z_1.\end{aligned}$$

Proceeding as in the previous case, we solve the phase plane equation,

$$\frac{dz_2}{dz_1} = \frac{\frac{ca}{b}z_1}{-\frac{br}{c}z_2},$$

for implicit solutions,

$$\frac{ca}{b} \frac{z_1^2}{2} + \frac{br}{c} \frac{z_2^2}{2} = C,$$

which correspond with ellipses and consequently orbital stability. Just as in the case of the pendulum equation, these orbits correspond with periodic behavior. \triangle

3.2 Stability and Eigenvalues

A more general (though less precise) method for studying stability regards the consideration of eigenvalues. In order to see how eigenvalues arise naturally in the analysis, consider again Example 3.1.

Example 3.1 Revisited. Consider again the second linearized equation from Example 3.1,

$$\begin{aligned}\frac{dy_1}{dt} &= y_2 \\ \frac{dy_2}{dt} &= \frac{g}{l}y_1.\end{aligned}\tag{3.6}$$

Recall that we can solve linear autonomous systems such as (3.6) by substitution. That is, writing $y_2 = \frac{dy_1}{dt}$ from the first equation and substituting it into the second, we have the second order equation

$$\frac{d^2y_1}{dt^2} = \frac{g}{l}y_1.$$

Homogeneous constant coefficient equations can be solved through the *ansatz* (guess) $y_1(t) = e^{rt}$, for which we have $r^2 - \frac{g}{l} = 0$, or $r = \pm\sqrt{\frac{g}{l}}$. According to standard ODE theory, we conclude that any solution $y_1(t)$ must have the form

$$y_1(t) = C_1e^{\sqrt{\frac{g}{l}}t} + C_2e^{-\sqrt{\frac{g}{l}}t}.$$

In the case of stability, $y_1(t)$ is expected to approach 0 as t gets large (or in the case of orbital stability, at least remain near 0). Since one summand of $y_1(t)$ grows exponentially as t gets large, we conclude that the point under consideration, $(\pi, 0)$, is unstable.

How do eigenvalues get involved in this? Writing

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad \text{and} \quad A = \begin{pmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{pmatrix},$$

we can write (3.6) in the system form,

$$y' = Ay.\tag{3.7}$$

If we define matrix exponentiation through Taylor expansion,

$$e^{At} = I + At + \frac{1}{2}A^2t^2 + \frac{1}{3!}A^3t^3 + \dots,$$

then as in the case with single equations, we can conclude

$$y(t) = e^{At} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

is a solution to (3.7). (This assertion can be checked through direct term-by-term differentiation.) In the event that A is diagonal (which is not the case in our example), e^{At} is straightforward to evaluate. For

$$A = \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix},$$

we have

$$\begin{aligned} e^{At} &= I + At + \frac{1}{2}A^2t^2 + \frac{1}{3!}A^3t^3 + \dots \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix}t + \frac{1}{2}\begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix}\begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix}t^2 \\ &\quad + \frac{1}{6}\begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix}\begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix}\begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix}t^3 + \dots \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix}t + \frac{1}{2}\begin{pmatrix} a_1^2 & 0 \\ 0 & a_2^2 \end{pmatrix}t^2 + \frac{1}{6}\begin{pmatrix} a_1^3 & 0 \\ 0 & a_2^3 \end{pmatrix}t^3 + \dots \\ &= \begin{pmatrix} e^{a_1t} & 0 \\ 0 & e^{a_2t} \end{pmatrix}. \end{aligned}$$

In the event that A is not diagonal, we will proceed by choosing a change of basis that diagonalizes A . This is where eigenvalues begin to emerge. Recall that eigenvalues, μ , of the matrix A are scalar constants that satisfy $Av = \mu v$ for some vector v , which is referred to as the eigenvector associated with μ . Typically, an $n \times n$ matrix will have n linearly independent eigenvectors. Observe that in the event that μ is an eigenvalue of the matrix A , we have

$$(A - \mu)v = 0, v \neq 0 \Rightarrow \det(A - \mu) = 0.$$

(If $\det(A - \mu) \neq 0$, we would conclude that $v = 0$ by standard matrix inversion.) We compute eigenvalues, then, by solving the polynomial equation, $\det(A - \mu) = 0$. (The polynomial $D(\mu) := \det(A - \mu)$ is typically referred to as the *characteristic polynomial*.) In our case, that is with

$$A = \begin{pmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{pmatrix},$$

we have

$$D(\mu) = \det \begin{pmatrix} -\mu & 1 \\ \frac{g}{l} & -\mu \end{pmatrix} = \mu^2 - \frac{g}{l} = 0 \Rightarrow \mu = \pm \sqrt{\frac{g}{l}}.$$

We can determine the eigenvectors associated with eigenvalues by solving

$$Av = \mu v \Rightarrow \begin{pmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \pm \sqrt{\frac{g}{l}} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \Rightarrow v_2 = \pm \sqrt{\frac{g}{l}}v_1.$$

Observe, in particular, that though we have two equations, we only get one relation for each eigenvalue. This means that one component of v can be chosen (almost) arbitrarily, which corresponds with the observation that if you multiply an eigenvector by a constant, you will get another (linearly dependent) eigenvector. In this case, let's choose $v_1 = 1$ for each eigenvector (recall that we should have two), giving

$$V_1 = \begin{pmatrix} 1 \\ \sqrt{\frac{g}{l}} \end{pmatrix}, \quad V_2 = \begin{pmatrix} 1 \\ -\sqrt{\frac{g}{l}} \end{pmatrix}.$$

Finally, we are prepared to diagonalize A . A general procedure for diagonalizing a matrix is outlined in the following three steps.

1. For an $n \times n$ matrix A , find n linearly independent eigenvectors of A , V_1, V_2, \dots, V_n .
2. Form a matrix P that consists of V_1 as its first column, V_2 as its second column, etc., with finally V_n as its last column.
3. The matrix $P^{-1}AP$ will then be diagonal with diagonal entries the eigenvalues associated with V_1, V_2, \dots, V_n : $\mu_1, \mu_2, \dots, \mu_n$.

Remark on Steps 1–3. First, it is not always the case that a matrix will have n linearly independent eigenvectors, and in situations for which this is not the case, more work is required (in particular, instead of diagonalizing the matrix, we put it in Jordan canonical form). Under the assumption that Step 1 is possible, the validity of Steps 2 and 3 is straightforward. If P is the matrix of eigenvectors, then

$$AP = (\mu_1 V_1 : \mu_2 V_2 : \dots : \mu_n V_n);$$

that is, the matrix containing as its k^{th} column the vector $\mu_k V_k$. Multiplying on the left by P^{-1} , which must exist if the V_k are all linearly independent, we have

$$P^{-1}(\mu_1 V_1 : \mu_2 V_2 : \dots : \mu_n V_n) = \begin{pmatrix} \mu_1 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \mu_n \end{pmatrix}.$$

In this last calculation, we are almost computing $P^{-1}P$, which would yield the identity matrix.

Returning to our example, we have

$$P = \begin{pmatrix} 1 & 1 \\ \sqrt{g/l} & -\sqrt{g/l} \end{pmatrix} \Rightarrow P^{-1} = \sqrt{\frac{l}{4g}} \begin{pmatrix} -\sqrt{g/l} & -1 \\ -\sqrt{g/l} & 1 \end{pmatrix},$$

with which we compute

$$\begin{aligned} D = P^{-1}AP &= \sqrt{\frac{l}{4g}} \begin{pmatrix} -\sqrt{g/l} & -1 \\ -\sqrt{g/l} & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ g/l & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ \sqrt{g/l} & -\sqrt{g/l} \end{pmatrix} \\ &= \begin{pmatrix} -\frac{1}{2} & -\frac{1}{2}\sqrt{\frac{l}{g}} \\ -\frac{1}{2} & \frac{1}{2}\sqrt{\frac{l}{g}} \end{pmatrix} \begin{pmatrix} \sqrt{g/l} & -\sqrt{g/l} \\ g/l & g/l \end{pmatrix} = \begin{pmatrix} -\sqrt{g/l} & 0 \\ 0 & +\sqrt{g/l} \end{pmatrix}. \end{aligned}$$

Observe now that we can write A in terms of its diagonalization, as $A = PDP^{-1}$. We can compute, then

$$\begin{aligned} e^{At} &= I + At + \frac{1}{2}A^2t^2 + \frac{1}{3!}A^3t^3 + \dots, \\ &= I + (PDP^{-1})t + \frac{1}{2}(PDP^{-1})^2 + \frac{1}{3!}(PDP^{-1})^3 + \dots \\ &= I + (PDP^{-1})t + \frac{1}{2}(PDP^{-1}PDP^{-1}) + \frac{1}{3!}(PDP^{-1}PDP^{-1}PDP^{-1}) + \dots \\ &= P(I + Dt + \frac{1}{2}D^2 + \frac{1}{3!}D^3 + \dots)P^{-1} \\ &= Pe^{Dt}P^{-1}. \end{aligned}$$

Consequently, we can write the solution of our example system (3.6) as

$$\begin{aligned}
y(t) &= e^{At} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = P e^{Dt} P^{-1} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 1 \\ \sqrt{\frac{g}{l}} & -\sqrt{\frac{g}{l}} \end{pmatrix} \begin{pmatrix} e^{-\sqrt{\frac{g}{l}}t} & 0 \\ 0 & e^{\sqrt{\frac{g}{l}}t} \end{pmatrix} \cdot \sqrt{\frac{l}{4g}} \begin{pmatrix} -\sqrt{\frac{g}{l}} & -1 \\ -\sqrt{\frac{g}{l}} & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \\
&= \begin{pmatrix} e^{-\sqrt{\frac{g}{l}}t} & e^{\sqrt{\frac{g}{l}}t} \\ \sqrt{\frac{g}{l}}e^{-\sqrt{\frac{g}{l}}t} & -\sqrt{\frac{g}{l}}e^{\sqrt{\frac{g}{l}}t} \end{pmatrix} \begin{pmatrix} -\frac{1}{2}c_1 - \frac{1}{2}\sqrt{\frac{l}{g}}c_2 \\ -\frac{1}{2}c_1 + \frac{1}{2}\sqrt{\frac{l}{g}}c_2 \end{pmatrix} \\
&= \begin{pmatrix} (-\frac{1}{2}c_1 - \frac{1}{2}\sqrt{\frac{l}{g}}c_2)e^{-\sqrt{\frac{g}{l}}t} + (-\frac{1}{2}c_1 + \frac{1}{2}\sqrt{\frac{l}{g}}c_2)e^{\sqrt{\frac{g}{l}}t} \\ (-\frac{1}{2}c_1 - \frac{1}{2}\sqrt{\frac{l}{g}}c_2)\sqrt{\frac{g}{l}}e^{-\sqrt{\frac{g}{l}}t} + (-\frac{1}{2}c_1 + \frac{1}{2}\sqrt{\frac{l}{g}}c_2) - \sqrt{\frac{g}{l}}e^{\sqrt{\frac{g}{l}}t} \end{pmatrix}.
\end{aligned}$$

Observing that in the absence of initial values the constants c_1 and c_2 remain arbitrary, we can recover the representation of $y_1(t)$ above by combining $C_1 = (-\frac{1}{2}c_1 - \frac{1}{2}\sqrt{\frac{l}{g}}c_2)$ and $C_2 = (-\frac{1}{2}c_1 + \frac{1}{2}\sqrt{\frac{l}{g}}c_2)$.

The primary observation I would like to make here is that the solution's rates of growth and decay are exactly the eigenvalues of the matrix A . If each of these is negative (or, in the case of complex eigenvalues, the real parts are negative) then we can conclude stability. If *any* of them are positive, we can conclude instability. In the event that a growth rate is 0 or purely imaginary, more analysis remains to be done. \triangle

Theorem 3.1. (*Linear ODE Stability*) For the linear first order system of ODE

$$y' = Ay, \quad y \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n},$$

the zero vector $y \equiv 0$ is stable or unstable as follows:

1. If all eigenvalues of A have nonpositive real parts, and all those with zero real parts are simple, then $y = 0$ is stable.
2. If and only if all eigenvalues of A have negative real parts, then $y = 0$ is asymptotically stable.
3. If one or more eigenvalues of A have a positive real part, then $y = 0$ is unstable.

Definition. For the general autonomous ODE,

$$y' = f(y), \quad y, f \in \mathbb{R}^n, \quad (3.8)$$

with equilibrium point y_e , if the linearization of (3.8) about y_e yields a linear equation for which the zero vector is stable, then we refer to y_e as linearly stable.

The eigenvalue approach to stability is clearly applicable to systems of any order.

Example 3.3. The simplest epidemic model is the SIR model,

$$\begin{aligned}
\frac{dS}{dt} &= -aSI \\
\frac{dI}{dt} &= aSI - bI \\
\frac{dR}{dt} &= bI,
\end{aligned}$$

where $S(t)$ represents the *susceptible* population, $I(t)$ represents the *infected* population, and $R(t)$ represents the *removed* population. Clearly, any point (S_e, I_e, R_e) for which $I_e = 0$ is an equilibrium point for this model. (And these are the only equilibrium points.) In order to study the stability of such points, we define the perturbation variables through

$$\begin{aligned}
S &= S_e + x \\
I &= 0 + y \\
R &= R_e + z.
\end{aligned}$$

Substituting into the SIR model, we find

$$\begin{aligned}\frac{dx}{dt} &= -a(S_e + x)y \\ \frac{dy}{dt} &= a(S_e + x)y - by \\ \frac{dz}{dt} &= by.\end{aligned}$$

Dropping the nonlinear terms we have the *linearized equations*

$$\begin{aligned}\frac{dx}{dt} &= -aS_e y \\ \frac{dy}{dt} &= (aS_e - b)y \\ \frac{dz}{dt} &= by.\end{aligned}$$

The system matrix A is given by

$$A = \begin{pmatrix} 0 & -aS_e & 0 \\ 0 & (aS_e - b) & 0 \\ 0 & b & 0 \end{pmatrix},$$

with eigenvalues

$$\det \begin{pmatrix} -\mu & -aS_e & 0 \\ 0 & (aS_e - b) - \mu & 0 \\ 0 & b & -\mu \end{pmatrix} = \mu^2((aS_e - b) - \mu) = 0 \Rightarrow \mu = 0, 0, aS_e - b.$$

In the event that $aS_e - b > 0$, we can conclude that this equilibrium point is unstable. This corresponds with the situation that the number of infectives grows faster than it dies off (by recovery or death). In this case, we would expect even a single infected person to cause an epidemic. In the case that $aS_e - b \leq 0$, we require a more detailed study. \triangle

3.3 Maximum Sustainable Yield

An issue closely related to stability is *maximum sustainable yield*. The maximum sustainable yield for a population of, say, fish, is the maximum number that can be harvested without killing off the population.

Example 3.4. Suppose that in the absence of fishermen the population of fish, $p(t)$, in a certain lake follows a logistic model, and that fishing yield is added as a percent of population. Determine the maximum sustainable yield for this population of fish and describe what will happen to the fish population if the maximum sustainable yield is harvested.

Keeping in mind that as long as we remain at non-zero stable equilibrium points the fish population will not die out, we begin by determining the equilibrium points for our model. Subtracting a percent harvest from the logistic model, we have

$$\frac{dp}{dt} = rp\left(1 - \frac{p}{K}\right) - hp,$$

where r and K are as described in Example 2.3 and h is the population harvest rate. In order to find the equilibrium points for this equation, we simply solve

$$rp_e\left(1 - \frac{p_e}{K}\right) - hp_e = 0 \Rightarrow p_e = 0, K\left(1 - \frac{h}{r}\right).$$

Since population can be regarded as positive, we conclude that a necessary condition for any sustainable yield is $h < r$, which simply asserts that we must catch fewer fish than are born.

In order to apply the eigenvalue approach to the stability of these equilibrium points, we recall that the eigenvalue of a 1×1 matrix (a.k.a. a scalar constant) is just the constant itself. For the equilibrium points $p_e = 0$, we obtain our linearized equation simply by dropping all high order terms, and we find

$$\frac{dp}{dt} = (r - h)p,$$

for which the matrix A is the scalar $A = r - h$. But we have already argued that $r > h$, so $A > 0$, and we can conclude instability. For the second equilibrium point, we introduce the perturbation variable $x(t)$ through

$$p(t) = K\left(1 - \frac{h}{r}\right) + x(t),$$

for which we find

$$\frac{dx}{dt} = r\left(x + K\left(1 - \frac{h}{r}\right)\right)\left(1 - \frac{x + K\left(1 - \frac{h}{r}\right)}{K}\right) - h\left(x + K\left(1 - \frac{h}{r}\right)\right).$$

Dropping high order terms, we have

$$\frac{dx}{dt} = -(r - h)x,$$

for which $A = -(r - h) < 0$, and we conclude stability. (In the case of single equations, stability is more readily observed directly, through consideration of the sign of $\frac{dp}{dt}$ for p above and below the equilibrium point (see the discussion of the logistic equation in Section 2.3), but it's instructive to see how the general approach works.) We conclude that so long as $h < r$, $p_e = K\left(1 - \frac{h}{r}\right)$ is an equilibrium point.

Finally, we choose our harvest rate h to maximize the yield, defined by

$$Y(h) = p_e h = hK\left(1 - \frac{h}{r}\right).$$

Maximizing in the usual way through differentiation, we have

$$Y'(h) = K\left(1 - \frac{h}{r}\right) - \frac{hK}{r} = 0 \Rightarrow h = \frac{r}{2}.$$

For this rate, our harvest is $\frac{r}{2}K\left(1 - \frac{r/2}{r}\right) = \frac{rK}{4}$, and the fish population approaches its equilibrium point $K\left(1 - \frac{r/2}{r}\right) = \frac{K}{2}$. \triangle

3.4 Uniqueness Theory

Example 3.5. Consider the following problem: Ignoring air resistance, determine an exact form for the time at which an object launched vertically from a height h with velocity v strikes the earth. According to Newton's second law of motion, the height of the object $y(t)$ can be described through,

$$y(t) = -g\frac{t^2}{2} + vt + h.$$

Setting $y(t) = 0$, we find,

$$-gt^2/(2h) + vt/h + 1 = 0,$$

with solution,

$$t = \frac{-v \pm \sqrt{v^2 + 2gh}}{-g}.$$

While we know that there is only one time at which the object can strike the ground, our model gives us two different times. This is a problem of uniqueness. (In this case, the resolution is straightforward: taking $-$ makes $t > 0$ and corresponds with the time we are looking for; taking $+$ makes $t < 0$ and corresponds with the object's trajectory being traced backward in time along its parabolic arc to the ground.) \triangle

Though the question of uniqueness arises in every type of equation—algebraic, differential, integral, integrodifferential, stochastic, etc.—we will only develop a (relatively) full theory in the case of ordinary differential equations.

Example 3.6. Consider the ordinary differential equation,

$$\frac{dy}{dt} = y^{2/3}; \quad y(0) = 0.$$

Solving by separation of variables, we find $y(t) = t^3/27$, which we compare with the following MATLAB script:

```

>>[t,y]=ode23(inline('y^(2/3)','t','y'),[0,5],0)
t =
0
0.0500
0.1000
0.1500
0.2000
0.2500
0.3000
0.3500
0.4000
0.4500
0.5000
y =
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0

```

According to MATLAB, the solution is $y(t) = 0$ for all t , and indeed it is straightforward to check that this is a valid solution to the equation. We find, in fact, that for any $c > 0$, the function $y(t)$ given as

$$y(t) = \begin{cases} \frac{(t-c)^3}{27}, & t \geq c \\ 0, & t \leq c \end{cases}$$

satisfies this equation. In practice, this is the fundamental issue with uniqueness: If our model does not have a unique solution, we don't know whether or not the solution MATLAB (or alternative software) gives us is the one that corresponds with the phenomenon we're modeling. \triangle

Two critical questions are apparent: 1. When can we insure that this problem won't arise (that solutions are unique)? and 2. In the case of nonuniqueness, can we develop a theory that selects the correct solution? The second of these questions can only be answered in the context of the phenomenon we're modeling. For example, in Example 3.5, we selected $t > 0$ because we were trying to predict a future time, and only one solution satisfied $t > 0$. As we observed, however, the other solution answered a different question that might have been posed: how long ago would the object have had to leave the ground to get to height h ? Fortunately, for the first of our two questions—at least in the case of ODE—we have a definitive general theorem.

Theorem 3.2. (*ODE Uniqueness*) Let $f(t, y) = (f_1(t, y), f_2(t, y), \dots, f_n(t, y))^{tr}$ be a vector function whose components are each continuous in both t and y in some neighborhood $a \leq t \leq b$ and $a_1 \leq y_1 \leq b_1, a_2 \leq y_2 \leq b_2, \dots, a_n \leq y_n \leq b_n$ and whose partial derivatives $\partial_{y_l} f_k(t, y)$ are continuous in both t and y in the same neighborhoods for each $l, k = 1, \dots, n$. Then given any initial point $(t_0, y_0) \in \mathbb{R} \times \mathbb{R}^n$ such that $a < t_0 < b$ and $a_k < y_{0_k} < b_k$ for all $k = 1, \dots, n$, any solution to

$$\frac{dy}{dt} = f(t, y); \quad y(t_0) = y_0$$

is unique on the neighborhood of continuity.

Example 3.6 continued. Notice that our equation from Example 3.6 better not satisfy the conditions of Theorem 3.2. In this case, $f(t, y) = y^{2/3}$, which is continuous in both t (trivially) and y . Computing $\partial_y f(t, y) = \frac{2}{3}y^{-1/3}$, we see that the y -derivative of f is not continuous at the initial value $y = 0$. \triangle

Example 3.7. Consider again the Lotka–Volterra predator–prey model, which we can re-write in the notation of Theorem 3.2 as ($y_1 = x$, $y_2 = y$),

$$\begin{aligned}\frac{dy_1}{dt} &= ay_1 - by_1y_2; & y_1(t_0) &= y_{0_1} \\ \frac{dy_2}{dt} &= -ry_2 + cy_1y_2; & y_2(t_0) &= y_{0_2}.\end{aligned}$$

In this case, the vector $f(t, y)$ is

$$\begin{pmatrix} f_1(t, y_1, y_2) \\ f_2(t, y_1, y_2) \end{pmatrix} = \begin{pmatrix} ay_1 - by_1y_2 \\ -ry_2 + cy_1y_2 \end{pmatrix}.$$

As polynomials, $f_1, f_2, \partial_{y_1} f_1, \partial_{y_2} f_1, \partial_{y_1} f_2$, and $\partial_{y_2} f_2$ must all be continuous for all t, y_1 , and y_2 , so any solution we find to these equations must be unique. \triangle

Idea of the uniqueness proof. Before proceeding with a general proof of Theorem 3.2, we will work through the idea of the proof in the case of a concrete example. Consider the ODE

$$\frac{dy}{dt} = y^2; \quad y(0) = 1, \tag{3.9}$$

and suppose we want to establish uniqueness on the intervals $a \leq t \leq b$ and $a_1 \leq y \leq b_1$, with $0 \in (a, b)$ and $1 \in (a_1, b_1)$. We begin by supposing that $y_1(t)$ and $y_2(t)$ are both solutions to (3.9) and defining the squared difference between them as a variable,

$$E(t) := (y_1(t) - y_2(t))^2.$$

Our goal becomes to show that $E(t) \equiv 0$; that is, that $y_1(t)$ and $y_2(t)$ must necessarily be the same function. Computing directly, we have

$$\begin{aligned}\frac{dE}{dt} &= 2(y_1(t) - y_2(t))\left(\frac{dy_1}{dt} - \frac{dy_2}{dt}\right) \\ &= 2(y_1(t) - y_2(t))(y_1(t)^2 - y_2(t)^2) \\ &= 2(y_1(t) - y_2(t))(y_1(t) - y_2(t))(y_1(t) + y_2(t)) \\ &= 2(y_1(t) - y_2(t))^2(y_1(t) + y_2(t)) \\ &= 2E(t)(y_1(t) + y_2(t)).\end{aligned}$$

Since y_1 and y_2 are both assumed less than b_1 , we conclude the differential inequality

$$\frac{dE}{dt} \leq 2E(t)(2b_1),$$

which upon multiplication by the (non-negative) *integrating factor* $e^{-4b_1 t}$ can be written as

$$\frac{d}{dt}[e^{-4b_1 t} E(t)] \leq 0.$$

Integrating, we have

$$\int_0^t \frac{d}{ds}[e^{-4b_1 s} E(s)] ds = e^{-4b_1 s} E(s) \Big|_0^t = e^{-4b_1 t} E(t) - E(0) \leq 0.$$

Recalling that $y_1(0) = y_2(0) = 1$, we observe that $E(0) = 0$ and consequently $E(t) \leq 0$. But $E(t) \geq 0$ by definition, so that we can conclude that $E(t) = 0$. \square

Proof of Theorem 3.2. In order to restrict the tools of this proof to a theorem that should be familiar to most students, we will carry it out only in the case of a single equation. The extension to systems is almost identical, only requiring a more general form of the Mean Value Theorem.

We begin as before by letting $y_1(t)$ and $y_2(t)$ represent two solutions of the ODE

$$\frac{dy}{dt} = f(t, y); \quad y(t_0) = y_0.$$

Again, we define the squared difference between $y_1(t)$ and $y_2(t)$ as $E(t) := (y_1(t) - y_2(t))^2$. Computing directly, we have now

$$\begin{aligned} \frac{dE}{dt} &= 2(y_1(t) - y_2(t))\left(\frac{dy_1}{dt} - \frac{dy_2}{dt}\right) \\ &= 2(y_1(t) - y_2(t))(f(t, y_1) - f(t, y_2)). \end{aligned}$$

At this point, we need to employ the Mean Value Theorem (see Appendix A), which asserts in this context that for each t there exists some number $c \in [y_1, y_2]$ so that

$$f'(c) = \frac{f(t, y_1) - f(t, y_2)}{y_1 - y_2}, \quad \text{or} \quad f(t, y_1) - f(t, y_2) = \partial_y f(t, c)(y_1 - y_2).$$

Since $\partial_y f$ is assumed continuous on the closed interval $t \in [a, b], y \in [a_1, b_1]$, the Extreme Value Theorem (see Appendix A) guarantees the existence of some constant L so that $|\partial_y f(t, y)| \leq L$ for all $t \in [a, b], y \in [a_1, b_1]$. We have, then, the so-called *Lipschitz inequality*,

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|. \quad (3.10)$$

We conclude that

$$\frac{dE}{dt} \leq 2|y_1(t) - y_2(t)|L|y_1(t) - y_2(t)| = 2LE(t),$$

from which we conclude exactly as above that $E(t) \equiv 0$. □

Remark on the Lipschitz Inequality. Often the ODE uniqueness theorem is stated under the assumption of the Lipschitz inequality (3.10). I have chosen to state it here in terms of the continuity of derivatives of f because that is typically an easier condition to check. Since continuity of derivatives implies the Lipschitz inequality, the Lipschitz formulation is more general.

3.5 Existence Theory

Existence theory is one of the most abstract topics in applied mathematics. The idea is to determine that a solution to some problem exists, even if the solution cannot be found.

Example 3.8. Prove that there exists a real solution to the algebraic equation

$$x^7 + 6x^4 + 3x + 9 = 0.$$

While actually finding a real solution to this equation is quite difficult, it's fairly easy to recognize that such a solution must exist. As x goes to $+\infty$, the left hand side becomes positive, while as x goes to $-\infty$ the left hand side becomes negative. Somewhere in between these two extremes, the left hand side must equal 0. In this way we have deduced that a solution exists without saying much of anything about the nature of the solution. (Mathematicians in general are notorious for doing just this sort of thing.) △

If we really wanted to ruin MATLAB's day, we could assign it the ODE

$$\frac{dy}{dt} = t^{-1}; \quad y(0) = 1.$$

Solving by direct integration, we see that $y(t) = \log t + C$, so that no value of C can match our initial data. (The current version of MATLAB simply crashes.) As with the case of uniqueness, we would like to

insure the existence of some solution before trying to solve the equation. Fortunately, we have the following theorem, due to Picard.

Theorem 3.3. (*ODE Existence*)⁷ Let $f(t, y) = (f_1(t, y), f_2(t, y), \dots, f_n(t, y))^{tr}$ be a vector function whose components are each continuous in both t and y in some neighborhood $a \leq t \leq b$ and $a_1 \leq y_1 \leq b_1, a_2 \leq y_2 \leq b_2, \dots, a_n \leq y_n \leq b_n$ and whose partial derivatives $\partial_{y_l} f_k(t, y)$ are continuous in both t and y in the same neighborhoods for each $l, k = 1, \dots, n$. Then given any initial point $(t_0, y_0) \in \mathbb{R} \times \mathbb{R}^n$ such that $a < t_0 < b$ and $a_k < y_{0k} < b_k$ for all $k = 1, \dots, n$, there exists a solution to the ODE

$$\frac{dy}{dt} = f(t, y); \quad y(t_0) = y_0 \quad (3.11)$$

for some domain $|t - t_0| < \tau$, where $\tau > 0$ may be extremely small. Moreover, the solution y is a continuous function of the independent variable t and of the parameters t_0 and y_0 .

Example 3.9. Consider the ODE

$$\frac{dy}{dt} = y^2; \quad y(0) = 1.$$

Since $f(t, y) = y^2$ is clearly continuous with continuous derivatives, Theorem 3.3 guarantees that a solution to this ODE exists. Notice particularly, however, that the interval of existence is not specified. To see exactly what this means, we solve the equation by separation of variables, to find

$$y(t) = \frac{1}{1-t},$$

from which we observe that though $f(y)$ and its derivatives are continuous for all t and y , existence is lost at $t = 1$. Referring to the statement of our theorem, we see that this statement is equivalent to saying that $\tau = 1$. Unfortunately, our general theorem does not specify τ for us a priori. \triangle

Idea of the proof of Theorem 3.3, single equations. Consider the ODE

$$\frac{dy}{dt} = y; \quad y(0) = 1.$$

Our goal here is to establish that a solution exists without even actually finding the solution. (Though if we accidentally stumble across a solution on our way, that's fine too.) We begin by simply integrating both sides, to obtain the *integral equation*

$$y(t) = 1 + \int_0^t y(s) ds.$$

(Unlike in the method of separation of variables, we have integrated both sides with respect to the same variable, t .) Next, we try to find a solution by an iteration. (Technically, *Picard Iteration*.) The idea here is that we guess at a solution, say $y_{\text{guess}}(t)$ and then use our integral equation to (hopefully) improve our guess through the calculation

$$y_{\text{new guess}}(t) = 1 + \int_0^t y_{\text{old guess}}(s) ds.$$

Typically, we call our first guess $y_0(t)$ and use the initial value: here, $y_0(t) = 1$. Our second guess, $y_1(t)$, becomes

$$y_1(t) = 1 + \int_0^t y_0(s) ds = 1 + \int_0^t 1 ds = 1 + t.$$

Similarly, we compute our next guess (iteration),

$$y_2(t) = 1 + \int_0^t y_1(s) ds = 1 + \int_0^t (1 + s) ds = 1 + t + \frac{t^2}{2}.$$

⁷The assumptions here are exactly the same as those for Theorem 3.2, so together Theorems 3.2 and 3.3 constitute a complete existence–uniqueness theory.

Proceeding similarly, we find that

$$y_n(t) = \sum_{k=0}^n \frac{t^k}{k!} \Rightarrow \lim_{n \rightarrow \infty} y_n(t) = \sum_{k=0}^{\infty} \frac{t^k}{k!},$$

and our candidate for a solution becomes $y(t) = \sum_{k=0}^{\infty} \frac{t^k}{k!}$, an infinite series amenable to such tests as the *integral test*, the *comparison test*, the *limit comparison test*, the *alternating series test*, and the *ratio test*. The last step is to use one of these tests to show that our candidate converges. We will use the ratio test, reviewed in Appendix A. Computing directly, we find

$$\lim_{k \rightarrow \infty} \frac{a_{k+1}}{a_k} = \frac{\frac{t^{k+1}}{(k+1)!}}{\frac{t^k}{k!}} = \frac{t^k t}{(k+1)k!} \cdot \frac{k!}{t^k} = \frac{t}{k+1} = 0, \text{ for all } t.$$

We conclude that $y(t)$ is indeed a solution. Observe that though we have developed a series representation for our solution, we have not found a closed form solution. (What is the closed form solution?) \triangle

Idea of the proof of Theorem 3.3, higher order equations and systems. We consider the ODE

$$\begin{aligned} y''(t) + y(t) &= 0; \\ y(0) &= 0; \quad y(1) = 1. \end{aligned} \tag{3.12}$$

In order to proceed as above and write (3.12) as an integral equation, we first write it in the notation of Theorem 4.2 by making the substitutions $y_1(t) = y(t)$ and $y_2(t) = y'(t)$:

$$\begin{aligned} \frac{dy_1}{dt} &= y_2; \quad y_1(0) = 0 \\ \frac{dy_2}{dt} &= -y_1; \quad y_2(0) = 1. \end{aligned}$$

(Notice that the assumptions of Theorem 3.3 clearly hold for this equation.) Integrating, we obtain the integral equations,

$$\begin{aligned} y_1(t) &= \int_0^t y_2(s) ds \\ y_2(t) &= 1 - \int_0^t y_1(s) ds. \end{aligned}$$

Our first three iterations become,

$$\begin{aligned} y_1(t)^{(1)} &= \int_0^t 1 ds = t \\ y_2(t)^{(1)} &= 1 - \int_0^t 0 ds = 1 \\ y_1(t)^{(2)} &= \int_0^t 1 ds = t \\ y_2(t)^{(2)} &= 1 - \int_0^t s ds = 1 - \frac{t^2}{2} \\ y_1(t)^{(3)} &= \int_0^t \left(1 - \frac{s^2}{2}\right) ds = t - \frac{t^3}{3!} \\ y_2(t)^{(3)} &= 1 - \int_0^t s ds = 1 - \frac{t^2}{2}. \end{aligned}$$

(By the way, I never said this was the world's most efficient algorithm.) Continuing, we find that

$$y(t) = y_1(t) = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \dots = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{t^{2k-1}}{(2k-1)!}.$$

Again, we can apply the ratio test to determine that this series converges (to what?). \square

Proof of Theorem 3.3. As with Theorem 3.2, we will only prove Theorem 3.3 in the case of single equations. The proof in the case of systems actually looks almost identical, where each statement is replaced by a vector generalization. I should mention at the outset that this is by far the most technically difficult proof of the semester. Not only is the argument itself fairly subtle, it involves a number of theorems from advanced calculus (e.g. M409).

Integrating equation (3.11), we obtain the integral equation

$$y(t) = y_0 + \int_{t_0}^t f(s, y(s)) ds.$$

Iterating exactly as in the examples above, we begin with y_0 and compute y_1, y_2, \dots according to

$$y_{n+1}(t) = y_0 + \int_{t_0}^t f(s, y_n(s)) ds; \quad n = 0, 1, 2, \dots$$

As both a useful calculation and a warmup for the argument to come, we will begin by estimating $\|y_1 - y_0\|$, where where $\|\cdot\|$ is defined similarly as in Theorem A.6 by

$$\|y(t)\| := \sup_{|t-t_0| < \tau} |y(t)|.$$

We compute

$$|y_1(t) - y_0| = \left| \int_{t_0}^t f(s, y_0) ds \right|.$$

Our theorem assumes that f is continuous on $s \in [t_0, t]$ and hence bounded, so there exists some constant M so that $\|f(s, y_0)\| \leq M$. We have, then

$$|y_1(t) - y_0| \leq \tau M.$$

Observing that the right-hand side is independent of t we can take supremum over t on both sides to obtain

$$\|y_1 - y_0\| \leq \tau M.$$

Finally, since we are at liberty to take τ as small as we like we will choose it so that $0 < \tau \leq \epsilon/M$, for some $\epsilon > 0$ to be chosen. In this way, we can insure that our new value y_1 remains in our domain of continuity of f .

We now want to look at the difference between two successive iterations and make sure the difference is getting smaller—that our iteration is actually making progress. For $|t - t_0| < \tau$, we have

$$\begin{aligned} |y_{n+1}(t) - y_n(t)| &= \left| \int_{t_0}^t (f(s, y_n(s)) - f(s, y_{n-1}(s))) ds \right| \\ &\leq \int_{t_0}^t L |y_n(s) - y_{n-1}(s)| ds \leq L\tau \sup_{|t-t_0| \leq \tau} |y_n(t) - y_{n-1}(t)|. \end{aligned}$$

Taking supremum over both sides (and observing that t has become a dummy variable on the right-hand side), we conclude

$$\|y_{n+1} - y_n\| \leq L\tau \|y_n - y_{n-1}\|,$$

Since τ is to be taken arbitrarily small, we can choose it to be as small as we like, and take $0 < \tau \leq L/2$. In this way, we have

$$\|y_{n+1} - y_n\| \leq \frac{1}{2} \|y_n - y_{n-1}\|.$$

We see that, indeed, on such a small interval of time our iterations are getting better. In fact, by carrying this argument back to our initial data, we find

$$\begin{aligned} \|y_{n+1} - y_n\| &\leq \frac{1}{2} \|y_n - y_{n-1}\| \leq \frac{1}{2} \cdot \frac{1}{2} \|y_{n-1} - y_{n-2}\| \\ &\leq \frac{1}{2^n} \|y_1 - y_0\| \leq \frac{\epsilon}{2^n}. \end{aligned}$$

In this way, we see that for $n > m$

$$\begin{aligned} \|y_n - y_m\| &= \left\| \sum_{k=m}^{n-1} (y_{k+1} - y_k) \right\| \leq \sum_{k=m}^{n-1} \|y_{k+1} - y_k\| \\ &\leq \epsilon \sum_{k=m}^{\infty} \frac{1}{2^k} = \frac{\epsilon}{2^{m-1}}. \end{aligned}$$

We conclude that

$$\lim_{n \rightarrow \infty} \|y_n - y_m\| = \lim_{n \rightarrow \infty} \frac{\epsilon}{2^{m-1}} = 0,$$

and thus by Cauchy's Convergence Condition (Theorem A.6) $y_n(t)$ converges to some function $y(t)$, which is our solution. \square

4 Solving Ordinary Differential Equations in MATLAB

MATLAB has an extensive library of functions for solving ordinary differential equations. In these notes, we will only consider the most rudimentary.

4.1 Finding Explicit Solutions

4.1.1 First Order Equations

Though MATLAB is primarily a numerics package, it can certainly solve straightforward differential equations symbolically.⁸ Suppose, for example, that we want to solve the first order differential equation

$$y'(x) = xy. \tag{4.1}$$

We can use MATLAB's built-in `dsolve()`. The input and output for solving this problem in MATLAB is given below.

```
>>y = dsolve('Dy = y*x','x')
y = C1*exp(1/2*x^2)
```

Notice in particular that MATLAB uses capital D to indicate the derivative and requires that the entire equation appear in single quotes. MATLAB takes t to be the independent variable by default, so here x must be explicitly specified as the independent variable. Alternatively, if you are going to use the same equation a number of times, you might choose to define it as a variable, say, `eqn1`.

```
>>eqn1 = 'Dy = y*x'
eqn1 =
Dy = y*x
>>y = dsolve(eqn1,'x')
y = C1*exp(1/2*x^2)
```

To solve an initial value problem, say, equation (4.1) with $y(1) = 1$, use

```
>>y = dsolve(eqn1,'y(1)=1','x')
y =
1/exp(1/2)*exp(1/2*x^2)
```

or

```
>>inits = 'y(1)=1';
>>y = dsolve(eqn1,inits,'x')
y =
1/exp(1/2)*exp(1/2*x^2)
```

⁸Actually, whenever you do symbolic manipulations in MATLAB what you're really doing is calling Maple.

Now that we've solved the ODE, suppose we want to plot the solution to get a rough idea of its behavior. We run immediately into two minor difficulties: (1) our expression for $y(x)$ isn't suited for array operations (`.*`, `./`, `.^`), and (2) y , as MATLAB returns it, is actually a symbol (a *symbolic object*). The first of these obstacles is straightforward to fix, using `vectorize()`. For the second, we employ the useful command `eval()`, which evaluates or executes text strings that constitute valid MATLAB commands. Hence, we can use

```
>>x = linspace(0,1,20);
>>z = eval(vectorize(y));
>>plot(x,z)
```

You may notice a subtle point here, that `eval()` evaluates strings (*character arrays*), and y , as we have defined it, is a symbolic object. However, `vectorize` converts symbolic objects into strings.

4.1.2 Second and Higher Order Equations

Suppose we want to solve and plot the solution to the second order equation

$$y''(x) + 8y'(x) + 2y(x) = \cos(x); \quad y(0) = 0, y'(0) = 1. \quad (4.2)$$

The following (more or less self-explanatory) MATLAB code suffices:

```
>>eqn2 = 'D2y + 8*Dy + 2*y = cos(x)';
>>inits2 = 'y(0)=0, Dy(0)=1';
>>y=dsolve(eqn2,inits2,'x')
y =
1/65*cos(x)+8/65*sin(x)+(-1/130+53/1820*14^(1/2))*exp((-4+14^(1/2))*x)
-1/1820*(53+14^(1/2))*14^(1/2)*exp(-(4+14^(1/2))*x)
>>z = eval(vectorize(y));
>>plot(x,z)
```

4.1.3 Systems

Suppose we want to solve and plot solutions to the system of three ordinary differential equations

$$\begin{aligned} x'(t) &= x(t) + 2y(t) - z(t) \\ y'(t) &= x(t) + z(t) \\ z'(t) &= 4x(t) - 4y(t) + 5z(t). \end{aligned} \quad (4.3)$$

First, to find a general solution, we proceed as in Section 4.1.1, except with each equation now braced in its own pair of (single) quotation marks:

```
>>[x,y,z]=dsolve('Dx=x+2*y-z','Dy=x+z','Dz=4*x-4*y+5*z')
x =
2*C1*exp(2*t)-2*C1*exp(t)-C2*exp(3*t)+2*C2*exp(2*t)-1/2*C3*exp(3*t)+1/2*C3*exp(t)
y =
2*C1*exp(t)-C1*exp(2*t)+C2*exp(3*t)-C2*exp(2*t)+1/2*C3*exp(3*t)-1/2*C3*exp(t)
z =
-4*C1*exp(2*t)+4*C1*exp(t)+4*C2*exp(3*t)-4*C2*exp(2*t)-C3*exp(t)+2*C3*exp(3*t)
```

(If you use MATLAB to check your work, keep in mind that its choice of constants $C1$, $C2$, and $C3$ probably won't correspond with your own. For example, you might have $C = -2C1 + 1/2C3$, so that the coefficients of $\exp(t)$ in the expression for x are combined. Fortunately, there is no such ambiguity when initial values are assigned.) Notice that since no independent variable was specified, MATLAB used its default, t . For an example in which the independent variable is specified, see Section 4.1.1. To solve an initial value problem, we simply define a set of initial values and add them at the end of our `dsolve()` command. Suppose we have $x(0) = 1$, $y(0) = 2$, and $z(0) = 3$. We have, then,

```

>>inits='x(0)=1,y(0)=2,z(0)=3';
>>[x,y,z]=dsolve('Dx=x+2*y-z','Dy=x+z','Dz=4*x-4*y+5*z',inits)
x =
6*exp(2*t)-5/2*exp(t)-5/2*exp(3*t)
y =
5/2*exp(t)-3*exp(2*t)+5/2*exp(3*t)
z =
-12*exp(2*t)+5*exp(t)+10*exp(3*t)

```

Finally, plotting this solution can be accomplished as in Section 4.1.1.

```

>>t=linspace(0,.5,25);
>>xx=eval(vectorize(x));
>>yy=eval(vectorize(y));
>>zz=eval(vectorize(z));
>>plot(t, xx, t, yy, t, zz)

```

The figure resulting from these commands is included as Figure 4.1.

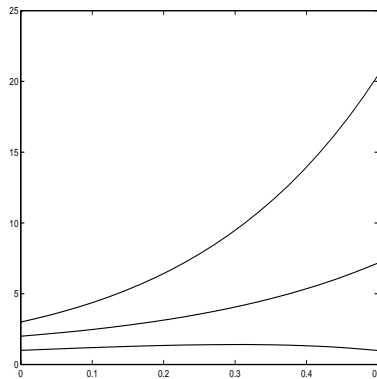


Figure 4.1: Solutions to equation (4.3).

4.2 Finding Numerical Solutions

MATLAB has a number of tools for numerically solving ordinary differential equations. We will focus on the main two, the built-in functions *ode23* and *ode45*, which implement versions of Runge–Kutta 2nd/3rd-order and Runge–Kutta 4th/5th-order, respectively.

4.2.1 First Order Equations

Suppose we want to numerically solve the first order ordinary differential equation $y'(x) = xy^2 + y$. First, we write an M-file, *firstode.m*, defining the function *yprime* as the derivative of y .⁹

```

function yprime = firstode(x,y);
% FIRSTODE: Computes yprime = x*y^2+y
yprime = x*y^2 + y;

```

Notice that all *firstode.m* does is take values x and y and return the value at the point (x, y) for the derivative $y'(x)$. A script for solving the ODE and plotting its solutions now takes the following form:

⁹Actually, for an equation this simple, we don't have to work as hard as we're going to work here, but I'm giving you an idea of things to come.

```

>>xspan = [0,.5];
>>y0 = 1;
>>[x,y]=ode23(@firstode,xspan,y0);
>>x
x =
    0
    0.0500
    0.1000
    0.1500
    0.2000
    0.2500
    0.3000
    0.3500
    0.4000
    0.4500
    0.5000
>>y
y =
    1.0000
    1.0526
    1.1111
    1.1765
    1.2500
    1.3333
    1.4286
    1.5384
    1.6666
    1.8181
    1.9999
>>plot(x,y)

```

Notice that *xspan* is the domain of x for which we're asking MATLAB to solve the equation, and $y_0 = 1$ means we're taking the initial value $y(0) = 1$. MATLAB solves the equation at discrete points and places the domain and range in vectors x and y . These are then easily manipulated, for example to plot the solution with $plot(x,y)$. Finally, observe that it is not the differential equation itself that goes in the function *ode23*, but rather the derivatives of the differential equation, which MATLAB assumes to be a first order system.

4.2.2 Second Order Equations

The first step in solving a second (or higher) order ordinary differential equation in MATLAB is to write the equation as a first order system. As an example, let's return to equation (4.2) from Section 4.1.2. Taking $y_1(x) = y(x)$ and $y_2(x) = y'(x)$, we have the system

$$\begin{aligned} y_1'(x) &= y_2(x) \\ y_2'(x) &= -8y_2(x) - 2y_1(x) + \cos(x) \end{aligned}$$

We now record the derivatives of this system as a function file. We have

```

function yprime = secondode(x,y);
%SECONDDODE: Computes the derivatives of y-1 and y-2,
%as a colum vector
yprime = [y(2); -8*y(2)-2*y(1)+cos(x)];

```

Observe that y_1 is stored as $y(1)$ and y_2 is stored as $y(2)$, each of which are column vectors. Additionally, *yprime* is a column vector, as is evident from the semicolon following the first appearance of $y(2)$. The MATLAB input and output for solving this ODE is given below.


```
>>xspan = [0,5];
>>y0 = [1;0];
>>[x,y]=ode23(@secondode,xspan,y0);

>>[x,y]

ans =
```

```
      0      1.0000      0
0.0001      1.0000     -0.0001
0.0005      1.0000     -0.0005
0.0025      1.0000     -0.0025
0.0124      0.9999     -0.0118
0.0296      0.9996     -0.0263
0.0531      0.9988     -0.0433
0.0827      0.9972     -0.0605
0.1185      0.9948     -0.0765
0.1613      0.9912     -0.0904
0.2113      0.9864     -0.1016
0.2613      0.9811     -0.1092
0.3113      0.9755     -0.1143
0.3613      0.9697     -0.1179
0.4113      0.9637     -0.1205
0.4613      0.9576     -0.1227
0.5000      0.9529     -0.1241
```

In the final expression above, the first column tabulates x values, while the second and third columns tabulate y_1 and y_2 ($y(1)$ and $y(2)$), or y and y' . MATLAB regards y as a matrix whose elements can be referred to as $y(m, n)$, where m refers to the row and n refers to the column. Here, y has two columns (y_1 and y_2) and 17 rows, one for each value of x . To get, for instance, the 4th entry of the vector $y(1)$, type $y(4,1)$ —4th row, 1st column. To refer to the entirety of y_1 , use $y(:,1)$, which MATLAB reads as *every* row, first column. Thus to plot y_1 versus y_2 we use `plot(y(:,1),y(:,2))`.

4.2.3 Solving Systems of ODE

Actually, we've already solved a system of ODE; the first thing we did in the previous example was convert our second order ODE into a first order system. As a second example, let's consider the Lorenz equations, which have some properties of equations arising in atmospheric, and whose solution has long served as an example for chaotic behavior. We have

$$\begin{aligned}\frac{dx}{dt} &= -\sigma x + \sigma y \\ \frac{dy}{dt} &= -y - xz \\ \frac{dz}{dt} &= -bz + xy - br,\end{aligned}\tag{4.4}$$

where for the purposes of this example, we will take $\sigma = 10$, $b = 8/3$, and $r = 28$, as well as $x(0) = -8$, $y(0) = 8$, and $z(0) = 27$. The MATLAB M-file containing the Lorenz equations appears below.

```
function xprime = lorenz(t,x);
%LORENZ: Computes the derivatives involved in solving the
%Lorenz equations.
sig=10;
b=8/3;
r=28;
xprime=[-sig*x(1) + sig*x(2); -x(2) - x(1)*x(3); -b*x(3) + x(1)*x(2) - b*r];
```

If in the Command Window, we type

```
>>tspan=[0,50];  
>>[t,x]=ode45(@lorenz,tspan,x0);  
>>plot(x(:,1),x(:,3))
```

the famous “Lorenz strange attractor” is sketched. (See Figure 4.2.)

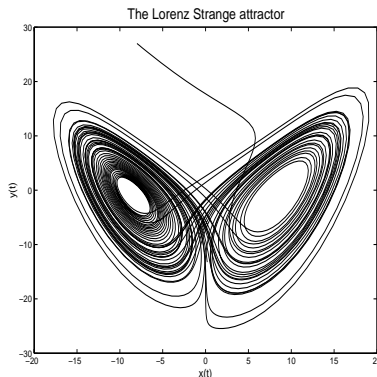


Figure 4.2: The Lorenz Strange Attractor

4.3 Laplace Transforms

One of the most useful tools in mathematics is the Laplace transform. MATLAB has built-in routines for computing both Laplace transforms and inverse Laplace transforms. For example, to compute the Laplace transform of $f(t) = t^2$, type simply

```
>>syms t;  
>>laplace(t^2)
```

In order to invert, say, $F(s) = \frac{1}{1+s}$, type

```
>>syms s;  
>>ilaplace(1/(1+s))
```

4.4 Boundary Value Problems

For various reasons of arguable merit most introductory courses on ordinary differential equations focus primarily on initial value problems (IVP’s). Another class of ODE’s that often arise in applications are boundary value problems (BVP’s). Consider, for example, the differential equation

$$\begin{aligned}y'' - 3y' + 2y &= 0 \\ y(0) &= 0 \\ y(1) &= 10,\end{aligned}$$

where our conditions $y(0) = 0$ and $y(1) = 10$ are specified on the boundary of the interval of interest $x \in [0, 1]$. (Though our solution will typically extend beyond this interval, the most common scenario in boundary value problems is the case in which we are only interested in values of the independent variable between the specified endpoints.) The first step in solving this type of equation is to write it as a first order system with $y_1 = y$ and $y_2 = y'$, for which we have

$$\begin{aligned}y_1' &= y_2 \\ y_2' &= -2y_1 + 3y_2.\end{aligned}$$

We record this system in the M-file *bvpexample.m*.

```

function yprime = bvpexample(t,y)
%BVPEXAMPLE: Differential equation for boundary value
%problem example.
yprime=[y(2); -2*y(1)+3*y(2)];

```

Next, we write the boundary conditions as the M-file *bc.m*, which records boundary *residues*.

```

function res=bc(y0,y1)
%BC: Evaluates the residue of the boundary condition
res=[y0(1);y1(1)-10];

```

By *residue*, we mean the left-hand side of the boundary condition once it has been set to 0. In this case, the second boundary condition is $y(1) = 10$, so its residue is $y(1) - 10$, which is recorded in the second component of the vector that *bc.m* returns. The variables *y0* and *y1* represent the solution at $x = 0$ and at $x = 1$ respectively, while the 1 in parentheses indicates the first component of the vector. In the event that the second boundary condition was $y'(1) = 10$, we would replace $y1(1) - 10$ with $y1(2) - 10$.

We are now in a position to begin solving the boundary value problem. In the following code, we first specify a grid of x values for MATLAB to solve on and an initial guess for the vector that would be given for an initial value problem $[y(0), y'(0)]$. (Of course, $y(0)$ is known, but $y'(0)$ must be a guess. Loosely speaking, MATLAB will solve a family of initial value problems, searching for one for which the boundary conditions are met.) We solve the boundary value problem with MATLAB's built-in solver *bvp4c*.

```

>>sol=bvpinit(linspace(0,1,25),[0 1]);
>>sol=bvp4c(@bvpexample,@bc,sol);
>>sol.x
ans =
Columns 1 through 9
0 0.0417 0.0833 0.1250 0.1667 0.2083 0.2500 0.2917 0.3333
Columns 10 through 18
0.3750 0.4167 0.4583 0.5000 0.5417 0.5833 0.6250 0.6667 0.7083
Columns 19 through 25
0.7500 0.7917 0.8333 0.8750 0.9167 0.9583 1.0000
>>sol.y
ans =
Columns 1 through 9
0 0.0950 0.2022 0.3230 0.4587 0.6108 0.7808 0.9706 1.1821
2.1410 2.4220 2.7315 3.0721 3.4467 3.8584 4.3106 4.8072 5.3521
Columns 10 through 18
1.4173 1.6787 1.9686 2.2899 2.6455 3.0386 3.4728 3.9521 4.4805
5.9497 6.6050 7.3230 8.1096 8.9710 9.9138 10.9455 12.0742 13.3084
Columns 19 through 25
5.0627 5.7037 6.4090 7.1845 8.0367 8.9726 9.9999
14.6578 16.1327 17.7443 19.5049 21.4277 23.5274 25.8196

```

We observe that in this case MATLAB returns the solution as a structure whose first component *sol.x* simply contains the x values we specified. The second component of the structure *sol* is *sol.y*, which is a matrix containing as its first row values of $y(x)$ at the x grid points we specified, and as its second row the corresponding values of $y'(x)$.

4.5 Event Location

Typically, the ODE solvers in MATLAB terminate after solving the ODE over a specified domain of the independent variable (the range we have referred to above as *xspan* or *tspan*). In applications, however, we often would like to stop the solution at a particular value of the dependent variable (for example, when an object fired from the ground reaches its maximum height or when a population crosses some threshold value). As an example, suppose we would like to determine the period of the pendulum from Example 3.1.

Since we do not know the appropriate time interval (in fact, that's what we're trying to determine), we would like to specify that MATLAB solve the equation until the pendulum swings through some specified fraction of its complete cycle and to give the time this took. In our case, we will record the time it takes the pendulum to reach the bottom of its arc, and multiply this by 4 to arrive at the pendulum's period. (In this way, the event is independent of the pendulum's initial conditions.) Our pendulum equation

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} \sin \theta$$

is stored in *pendode.m* with $l = 1$ (see Example 3.1). In addition to this file, we write an *events* file *pendevent.m* that specifies the event we are looking for.

```
function [lookfor stop direction]=pendevent(t,x)
%PENDEVENT: MATLAB function M-file that contains the event
%that our pendulum reaches its center point from the right
lookfor = x(1) ; %Searches for this expression set to 0
stop = 1; %Stop when event is located
direction = -1; %Specify direction of motion at event
```

In *pendevent.m*, the line *lookfor=x(1)* specifies that MATLAB should look for the event $x(1) = 0$ (that is, $x(t) = 0$). (If we wanted to look for the event $x(t) = 1$, we would use *lookfor=x(1)-1*.) The line *stop=1* instructs MATLAB to stop solving when the event is located, and the command *direction=-1* instructs MATLAB to only accept events for which $x(2)$ (that is, x') is negative (if the pendulum starts to the right of center, it will be moving in the negative direction the first time it reaches the center point).

We can now solve the ODE up until the time our pendulum reaches the center point with the following commands issued in the Command Window:

```
>>options=odeset('Events',@pendevent);
>>x0=[pi/4 0];
>>[t, x, te, xe, ie]=ode45(@pendode, [0, 10], x0, options);
>>te
te =
0.5215
>>xe
xe =
-0.0000 -2.3981
```

Here, x_0 is a vector of initial data, for which we have chosen that the pendulum begin with angle $\pi/4$ and with no initial velocity. The command *ode45()* returns a vector of times t , a matrix of dependent variables x , the time at which the event occurred, te , the values of x when the event occurred, xe , and finally the index when the event occurred, ie . In this case, we see that the event occurred at time $t = .5215$, and consequently the period is $P = 2.086$ (within numerical errors). Though the exact period of the pendulum is difficult to analyze numerically, it is not difficult to show through the small angle approximation $\sin \theta \cong \theta$ that for θ small the period of the pendulum is approximately $P = 2\pi\sqrt{\frac{l}{g}}$, which in our case gives $P = 2.001$. (While the small angle approximation gives a period independent of θ , the period of a pendulum does depend on θ .)

5 Numerical Methods

Though we can solve ODE on MATLAB without any knowledge of the numerical methods it employs, it's often useful to understand the basic underlying principles. In order to gain a basic understanding of how ODE solvers work in general, we recall the development of Euler's method.

5.1 Euler's Method

Consider the first order ODE

$$y' = f(t, y); \quad y(0) = y_0.$$

Recalling the definition of derivative,

$$y'(t) = \lim_{h \rightarrow 0} \frac{y(t+h) - y(t)}{h},$$

we suspect that for h sufficiently small,

$$y'(t) \cong \frac{y(t+h) - y(t)}{h}.$$

If this is the case, then we have the approximate equation

$$\frac{y(t+h) - y(t)}{h} = f(t, y(t)),$$

which we can rearrange as

$$y(t+h) = y(t) + hf(t, y(t)).$$

We now proceed by taking h small and computing $y(t)$ iteratively. For example, taking $h = .1$, and beginning with the known initial point $y(0) = y_0$, we have the sequence of iterations

$$\begin{aligned} y(.1) &= y(0) + .1f(0, y(0)) \\ y(.2) &= y(.1) + .1f(.1, y(.1)) \\ &\vdots \end{aligned}$$

Generally, we describe this iteration through the iterative relation

$$y(t_{n+1}) = y(t_n) + \Delta t f(t_n, y(t_n)).$$

We refer to processes in which $y(t_{n+1})$ depends only on t_n and $y(t_n)$ (with Δt assumed fixed) as *single-step* solvers. In the event that our solver depends on additional previous times, we refer to the solver as a *multistep* solver.

6 Advanced ODE Solvers

In addition to the ODE solvers *ode23* and *ode45*, which are both based on the Runge–Kutta scheme, MATLAB has several additional solvers, listed below along with MATLAB's help-file suggestions regarding when to use them.

- Multistep solvers
 - *ode113*. If using stringent error tolerances or solving a computationally intensive ODE file.
- Stiff problems (see discussion below)
 - *ode15s*. If *ode45* is slow because the problem is stiff.
 - *ode23s*. If using crude error tolerances to solve stiff systems and the mass matrix is constant.
 - *ode23t*. If the problem is only moderately stiff and you need a solution without numerical damping.
 - *ode23tb*. If using crude error tolerances to solve stiff systems.

6.1 Stiff ODE

By a *stiff* ODE we mean an ODE for which numerical errors compound dramatically over time. For example, consider the ODE

$$y' = -100y + 100t + 1; \quad y(0) = 1.$$

Since the dependent variable, y , in the equation is multiplied by 100, small errors in our approximation will tend to become magnified. In general, we must take considerably smaller steps in time to solve stiff ODE, and this can lengthen the time to solution dramatically. Often, solutions can be computed more efficiently using one of the solvers designed for stiff problems.

Fundamental Theorems

One of the most useful theorems from calculus is the Implicit Function Theorem, which addresses the question of existence of solutions to algebraic equations. Instead of stating its most general version here, we will state exactly the case we use.

Theorem A.1. (Implicit Function Theorem) Suppose the function $f(x_1, x_2, \dots, x_n)$ is C^1 in a neighborhood of the point (p_1, p_2, \dots, p_n) (the function is continuous at this point, and its derivatives with respect to each variable are also continuous at this point). Suppose additionally that

$$f(p_1, p_2, \dots, p_n) = 0$$

and

$$\partial_{x_1} f(p_1, p_2, \dots, p_n) \neq 0.$$

Then there exists a neighborhood N_p of (p_2, p_3, \dots, p_n) and a function $\phi : N_p \rightarrow \mathbb{R}$ so that

$$p_1 = \phi(p_2, p_3, \dots, p_n),$$

and for every $x \in N_p$,

$$f(\phi(x_2, x_3, \dots, x_n), x_2, x_3, \dots, x_n) = 0.$$

Theorem A.2. (Mean Value Theorem) Suppose $f(x)$ is a differentiable function on the interval $x \in [a, b]$. There there exists some number $c \in [a, b]$ so that

$$f'(c) = \frac{f(b) - f(a)}{b - a}.$$

Theorem A.3. (Extreme Value Theorem) Suppose $f(x)$ is a function continuous on a closed interval $x \in [a, b]$. Then $f(x)$ attains a bounded absolute maximum value $f(c)$ and a bounded absolute minimum value $f(d)$ at some numbers c and d in $[a, b]$.

Theorem A.4. (The Ratio Test) For the series $\sum_{k=1}^{\infty} a_k$, if $\lim_{k \rightarrow \infty} \left| \frac{a_{k+1}}{a_k} \right| = L < 1$, then the series is absolutely convergent (which means that not only does the series itself converge, but a series created by taking absolute values of the summands in the series also converges). On the other hand, if $\lim_{k \rightarrow \infty} \left| \frac{a_{k+1}}{a_k} \right| = L > 1$ or $\lim_{k \rightarrow \infty} \left| \frac{a_{k+1}}{a_k} \right| = \infty$ the series diverges. if $\lim_{k \rightarrow \infty} \left| \frac{a_{k+1}}{a_k} \right| = 1$, the ratio test is inconclusive.

Theorem A.5. (Cauchy's Convergence Condition) Let $\{a_n\}_{n=1}^{\infty}$ be a sequence of points and consider the limit $\lim_{n \rightarrow \infty} a_n$. A necessary and sufficient condition that this limit be convergent is that

$$\lim_{n > m \rightarrow \infty} |a_n - a_m| = 0.$$

Theorem A.6. (Cauchy's Convergence Condition for functions, in exactly the form we require) Let the series of functions $\{y_n(t)\}_{n=1}^{\infty}$ be defined for $t \in [a, b]$, and define $\| \cdot \|$ by the relation

$$\|y(t)\| := \sup_{t \in [a, b]} |y(t)|.$$

Then if

$$\lim_{n > m \rightarrow \infty} \|y_n(t) - y_m(t)\| = 0,$$

we have

$$\lim_{n \rightarrow \infty} y_n(t)$$

converges uniformly on $t \in [a, b]$.

Theorem A.7. (Taylor expansion with remainder) Suppose $f(x)$ and its first n derivatives are continuous for $x \in [a, b]$, and suppose the $(n + 1)$ st derivative $f^{(n+1)}(x)$ exists for $x \in (a, b)$. Then there is a value $X \in (a, b)$ so that

$$f(b) = f(a) + f'(a)(b - a) + \dots + \frac{f^{(n)}(a)}{n!}(b - a)^n + \frac{f^{(n+1)}(X)}{(n + 1)!}(b - a)^{n+1}.$$

Index

- boundary value problems, 42
- carrying capacity, 9
- characteristic polynomial, 26
- chemical reactions, 4
- critical points, 21
- dsolve, 37
- eigenvalues, 25
- equilibrium points, 21
- eval, 38
- event location, 43
- Extreme Value Theorem, 46
- Gompertz model, 9
- half saturation constants, 10
- Implicit Function Theorem, 46
- instability
 - saddle, 24
- integral curves, 23
- integral equation, 34
- inverse laplace, 42
- laplace, 42
- Laplace transforms, 42
- law of mass action, 5
- Lipschitz inequality, 33
- logistic model, 9
- Lotka-Volterra model, 9
- Malthusian model, 8
- maximum sustainable yield, 29
- Mean Value Theorem, 46
- multistep solver, 45
- ode, 39
- ode113(), 45
- ode15s(), 45
- ode23s(), 45
- ode23t(), 45
- ode23tb(), 45
- phase plane equation, 21
- phase portrait, 18
- Picard Iteration, 34
- population dynamics, 8
- single-step solvers, 45
- SIR model, 10
- stability
 - orbital, 23
- stiff ODE, 46
- trajectories, 23
- transition matrix, 11
- uniqueness theory, 30
- vectorize(), 38
- well-posedness, 17